Classificação Hierárquica da Atividade Económica das Empresas a partir de Texto da *Web*

por

Maria da Conceição da Silva Ferreira

Tese de Mestrado em Análise de Dados e Sistemas de Apoio à Decisão

Orientada por

Professor Doutor Pavel Bernard Brazdil

Co-orientada por

Professor Doutor Pedro José Ramos Moreira de Campos

Faculdade de Economia

Universidade do Porto

2012

Nota Biográfica

Conceição Ferreira nasceu a 30 de Novembro de 1981 em Viseu. Viveu a sua infância e adolescência em Castro Daire, vila do distrito de Viseu, onde concluiu a sua formação básica e secundária. Licenciou-se em Matemática – Área de Especialização em Matemática Aplicada pela Escola de Ciências da Universidade do Minho, em Braga, em 2007. A sua atividade profissional iniciou-se em 2009 no Instituto Nacional de Estatística (INE), em Lisboa. Em 2010 foi transferida para a Delegação Norte, na cidade do Porto, onde ainda se encontra atualmente, e desempenha funções de Técnica Superior de Estatística, no Serviço de Métodos Estatísticos.

Agradecimentos

Ao Professor Doutor Pavel Brazdil, orientador do presente trabalho, pela sua disponibilidade que sempre demonstrou, bem como pela revisão dos conteúdos e críticas construtivas, que permitiram a evolução do trabalho.

Ao co-orientador e também colega de trabalho Professor Doutor Pedro Campos pelo companheirismo e amizade presentes nas trocas de ideias havidas.

À minha mãe e restante família pelo constante apoio e conforto, sempre necessários para a saúde emocional.

Ao meu marido, Carlos Eduardo, por todo o carinho, amizade e compreensão, em todos os momentos.

Aos meus amigos pela sua amizade e palavras de incentivo, que fizeram com que nunca desistisse de prosseguir o caminho iniciado.

Aos meus colegas de mestrado pelo espírito de grupo que sempre insistiram em manter, importante para sentir que não estamos sozinhos.

Aos meus colegas de trabalho por toda a força e incentivo.

Motivação

O percurso agora a terminar, inicia-se com a necessidade quase inerente ao ser humano de querer saber sempre mais, ou seja, de não estagnar na aquisição de conhecimento. A decisão de obtenção de um grau de Mestre foi bastante ponderada e a área escolhida foi cuidadosamente pensada. A opção pelo mestrado em Análise de Dados e Sistemas de Apoio à Decisão, apareceu, primeiramente pela afinidade natural com a formação de base em Matemática Aplicada e, finalmente, pela crescente procura de ferramentas de Sistemas de Apoio à Decisão no meio empresarial, como forma de valorização e desenvolvimento dos recursos humanos e materiais das empresas.

O tema da presente dissertação foi eleito, em conjunto com os orientadores, pela sua atualidade e potencial interesse e aplicabilidade em contexto real, com as necessárias adaptações e ajustes.

Resumo

Os constantes avanços na tecnologia proporcionam a atuação das empresas em qualquer mercado mundial, independentemente da sua localização efetiva. Nos tempos que correm, a Internet contínua a revolucionar a comunicação e o acesso à informação. Assim, através da Internet, uma empresa pode partilhar com o mundo a sua área de negócio. Neste estudo, começa-se por criar uma coleção de documentos, em que cada um dos documentos representa a informação sobre a atividade económica de uma empresa, obtida a partir da respetiva página na Internet. Posteriormente, através de técnicas de *data mining/text mining*, pretende-se atribuir um dado documento a uma de várias categorias. As categorias representam algumas Secções e Divisões da Classificação Portuguesa das Atividades Económicas, Revisão 3 (CAE-Rev.3), e estão estruturadas hierarquicamente até ao nível da Divisão. Assim, o objetivo principal deste trabalho consiste na construção de classificadores, que permitam classificar documentos de texto em diferentes categorias organizadas hierarquicamente, segundo duas abordagens, a classificação local por nó pai e a classificação local por nível. A análise de similaridade entre os documentos com o descritivo das empresas e documentos com o descritivo das categorias é apontada como objetivo secundário, com o intuito de verificar se traz algum ganho face ao método de classificação. Os objetivos da tese foram atingidos, pois conseguiu-se construir um classificador (*Naive Bayes*) que, na classificação de documentos no primeiro nível, obteve uma medida de desempenho, microF1, na ordem dos 80%, considerando-se, no geral, bastante boa. Na classificação do segundo nível, o classificador construído considerando a abordagem classificação local por nó pai, mostrou-se com melhor desempenho do que o construído considerando a abordagem classificação local por nível em metade das categorias, nas restantes verificou-se o contrário. A análise de similaridade não revelou nenhum ganho face ao método de classificação.

Abstract

The constant advances in technology provide corporate action on any world market, regardless of their actual location. These days, the Internet continues to revolutionize communication and access to information. Thus, through the Internet, a company can share with the world their business area. This study begins by creating a collection of documents, where each document represents information on the economic activity of a company, obtained from the respective website. Later, through techniques of data mining / text mining, the aim is to assign a given document to one of several categories. The categories represent some Sections and Divisions of the Economic Activity (NACE Rev. 2), and are structured hierarchically to level Division. Thus, the main objective of this work is to construct classifiers that allow classifying text documents into different categories hierarchically organized, according to two approaches, the local classifier per parent node and the local classifier per level. The analysis of similarity between documents with the descriptive of the companies and documents with the descriptive of the categories is identified as a secondary objective, in order to verify if it brings any gain over the classification method. The objectives of the thesis were achieved, it was possible to build a classifier (*Naive Bayes*) that, in document classification at the first level, got a measure of performance, microF1, in the order of 80%, considered, in general, quite good. In the second level classification, the classifier constructed considering the local classifier per parent node approach, showed a better performance than the classifier built considering the local classifier per level approach in half of the categories, in the other half it was the opposite. The similarity analysis revealed no gain over the classification method.

Índice

Lista de Tabelas

Lista de Figuras

CAPÍTULO 1
Introdução

As empresas utilizam cada vez mais a Internet para disponibilizar aos utilizadores toda a informação sobre as respetivas áreas de negócio e serviços, tendo como principal objetivo a angariação de potenciais clientes, pela entrada e extensão a novos mercados.

O presente trabalho tem como objetivo a construção de classificadores, que permitam classificar documentos de texto em diferentes categorias organizadas hierarquicamente, aplicando técnicas de *data mining* / *text mining*, em particular, métodos de classificação de documentos de texto.

As técnicas mencionadas são cada vez mais utilizadas por empresas, como forma de maximizar a sua operacionalidade e, consequentemente, valorizar a própria organização. Alguns exemplos de aplicação prática destas técnicas são a classificação de *email* (por exemplo, arquivar por *spam*), a classificação de documentos (por exemplo, organizar por tipo), entre outros.

Os documentos de texto considerados neste estudo referem-se ao descritivo da atividade económica de algumas empresas, obtido a partir da respetiva página na Internet, *website* ou diretórios existentes. Após a recolha dos dados (documentos de texto) é criado um corpus e é efetuado o seu pré-processamento com o uso de técnicas de *text mining*.

Todas as empresas têm a sua atividade classificada, segundo a Classificação Portuguesa das Atividades Económicas, Revisão 3 (CAE-Rev.3), o que permite que se agrupem de acordo com a mesma. Assim, as categorias usadas para classificar os documentos são representativas do código CAE-Rev.3.

As empresas podem ter a sua atividade económica classificada em mais do que um código CAE-Rev.3. Neste estudo, considera-se apenas um código CAE-Rev.3, em cada uma das empresas, relativo à sua atividade principal.

O problema proposto é um problema de classificação hierárquica (as categorias estão organizadas hierarquicamente numa estrutura em árvore) e multi-classe (existem mais

do que duas categorias para classificar os documentos). A base de aprendizagem deste problema é a aprendizagem supervisionada no qual as classes estão previamente definidas. Cada documento deve inevitavelmente encaixar-se numa só categoria.

Apesar de a CAE-Rev.3 estar estruturada em 5 níveis (Secções, Divisões, Grupos, Classes e Subclasses), as categorias representativas da CAE-Rev.3 utilizadas para classificação, neste estudo, foram limitadas às Secções F, G, entre outras que foram agrupadas originando a categoria 'OUTRA'. Para além das Secções, consideraram-se as Divisões das Secções F e G. Portanto, no primeiro nível da hierarquia tem-se três categorias, enquanto no segundo nível da hierarquia tem-se seis categorias, Divisões 41, 42 e 43 da Secção F e Divisões 45, 46 e 47 da Secção G.

O objetivo principal deste trabalho consiste em classificar uma coleção de documentos de empresas, seguindo duas abordagens diferentes. A primeira é definida como classificação local por nó pai e a segunda é definida como classificação local por nível.

O classificador no primeiro nível é treinado para classificar os documentos numa de 3 categorias ('F', 'G' ou 'OUTRA'). Para obter esse classificador, é necessário gerar a matriz documentos por termos a partir do Corpus. Esta situação é idêntica nas duas abordagens. Ao nível da classificação no segundo nível é que são observadas diferenças. Na primeira abordagem (classificação local por nó pai), para distinguir as categorias '41', '42' ,'43', '45', '46' e '47' é preciso ter dois classificadores, um que será treinado apenas os exemplos de categoria 'F' e que classificará os documentos nas categorias '41', '42' ou '43'; e um segundo que será treinado apenas os exemplos de categoria 'G' e que classificará os documentos nas categorias '45', '46' ou '47'. Na segunda abordagem (classificação local por nível), para distinguir as categorias '41', '42' ,'43', '45', '46' e '47' é construído apenas um classificador que será treinado com os exemplos das categorias 'F' e 'G'.

Os objetivos secundários deste trabalho correspondem a analisar os efeitos do uso de documentos com e sem *stemming,* e classificar os documentos segundo a análise de similaridade entre documentos com a descrição das empresas e documentos com a descrição das categorias, considerando a medida de similaridade 'cosseno'.

Como principais resultados deste trabalho destacam-se a boa performance do classificador, *Naive Bayes*, construído para o primeiro nível, com uma taxa de acerto de aproximadamente 81%; a boa performance do classificador, também *Naive Bayes*, construído para o segundo nível considerando a classificação local por nó pai, no caso em que o nó pai é a categoria 'G', sendo que a taxa de acerto na classificação dos descendentes desta categoria ronda os 80%. A classificação dos descendentes da categoria 'F' obteve melhores resultados na abordagem classificação local por nível. A análise de similaridade não revelou ganhos face ao método de classificação.

Os classificadores foram construídos a partir de exemplos de documentos de texto, com o apoio de métodos de aprendizagem automática.

A ferramenta utilizada neste projeto foi o R, versão 2.14.0. O *software* R pode ser adquirido gratuitamente no *site* CRAN (*The Comprehensive R Archive Network*) em http://cran.r-project.org. Apesar de gratuito é uma ferramenta bastante poderosa, com boas capacidades a nível de programação e com um conjunto de *packages* em ascensão [2]. Como o R é uma linguagem de programação orientada a objetos o utilizador pode criar as suas próprias funções e rotinas para a análise e manipulação de dados.

A dissertação apresenta a descrição de todos os passos executados e os resultados obtidos em todo o processo de classificação de documentos de texto.

A dissertação encontra-se estruturada por capítulos. O primeiro capítulo corresponde à introdução. No segundo capítulo é feita uma revisão dos métodos de classificação de documentos, onde são abordadas todas as matérias envolvidas nesta tese. Começa-se por descrever as técnicas de *data mining* e *text mining*, as tarefas de pré-processamento utilizadas, metodologia de classificação de documentos, em particular a classificação hierárquica. Ainda neste capítulo são abordados os classificadores utilizados neste estudo, bem como os métodos de avaliação da performance dos mesmos. É ainda definida a análise de similaridade entre documentos. No capítulo 3 descreve-se o caso de estudo e todas as temáticas à sua resolução. No capítulo 4 são apresentados os resultados dos objetivos propostos e é feita uma discussão dos mesmos. No capítulo 5 são apresentadas as conclusões deste estudo bem como sugestões para trabalhos futuros.

CAPÍTULO 2

Métodos de Classificação de Documentos

Neste capítulo serão abordados conceitos e metodologias imprescindíveis para o entendimento deste trabalho. Começa-se por abordar a metodologia *data mining* e uma das aplicações mais comuns desta metodologia, a classificação. Neste seguimento será abordado *text mining* e algumas tarefas de pré-processamento. Posteriormente é definida a classificação de documentos, em particular, a classificação hierárquica e multi-classe. Alguns algoritmos de classificação, como árvores de decisão, *k*-vizinhos mais próximos, *support vector machines*, redes neuronais e *Naive Bayes* serão abordados, bem como os métodos de avaliação da performance dos algoritmos. Finalmente referida a abordagem da análise de similaridade entre documentos.

2.1. *Data Mining*

A metodologia *data mining* refere-se à extração de conhecimento de grandes quantidades de dados. *Data mining* é vista como um passo essencial no processo de descoberta de conhecimento a partir de dados (*Knowledge Discovery from Data* - KDD) (Han e Kamber, 2006). A Figura 1 mostra as etapas do processo de KDD.

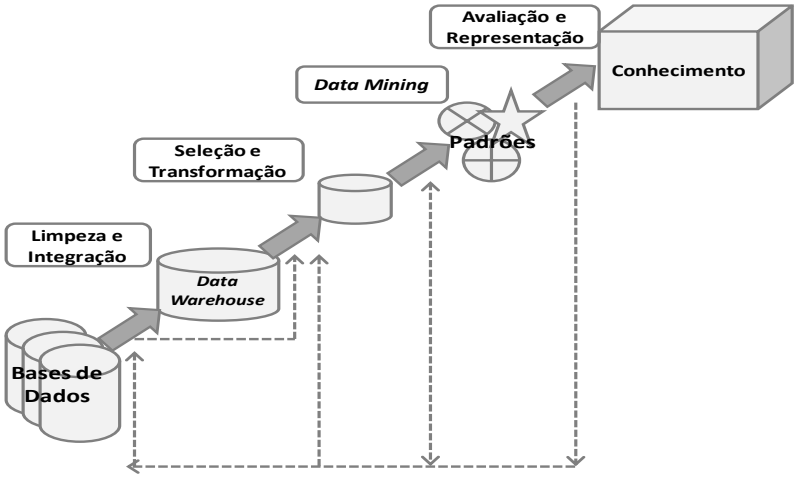


Figura 1: Processo de Descoberta de Conhecimento

Data mining utiliza técnicas de inteligência artificial com o objetivo de encontrar, de forma automática, padrões capazes de transformar os dados em informação útil.

Os dados estruturados são geralmente a fonte de um processo de *data mining*. Normalmente, a representação de dados estruturados é na forma de um quadro, em que as colunas são características de objetos armazenados numa tabela e as linhas são valores dessas características (Kantardzic, 2003). Na literatura de *data mining* é frequente usarem-se os termos amostras, instâncias, exemplos ou casos para as linhas e variáveis, atributos ou características para as colunas.

Os dados podem ser rotulados ou não rotulados. A Tabela 1 apresenta um exemplo de um conjunto de dados rotulados.

Tabela 1: Exemplo típico de um conjunto de dados rotulados em *data mining*

		Atributos			**Atributo Especial**
		A	**B**	**C**	**Categoria**
Instâncias	**1**	Falso	69	70	sim
	2	Verdadeiro	80	90	não
	…	…	…	…	…
	n	Verdadeiro	75	70	sim

Na presença de dados rotulados o objetivo consiste em utilizar os dados fornecidos para predizer o valor do atributo especial para novas instâncias. Este tipo de aprendizagem é conhecido como aprendizagem supervisionada. No caso de dados não rotulados o objetivo é simplesmente extrair o máximo de informação a partir dos dados disponíveis. A utilização deste tipo de dados é conhecida como aprendizagem não supervisionada (Bramer, 2007).

A automatização do processo de aprendizagem é conhecida como aprendizagem automática (*machine learning*). Em *data mining*, são usados algoritmos de aprendizagem automática para descobrir o conhecimento de grandes bases de dados (Mitchell, 1997). Os diferentes algoritmos variam nos seus objetivos, nos conjuntos de dados de treino disponíveis e nas estratégias de aprendizagem e de representação de dados (Kantardzic, 2003).

Em geral, as tarefas de *data mining* podem ser agrupadas em duas categorias: preditiva e descritiva. A preditiva corresponde a prever se um item pertence ou não a uma categoria. A descritiva corresponde, de um modo geral, a extrair padrões a partir dos exemplos (Awad *et al.*, 2009).

A classificação é uma das aplicações mais comuns de *data mining*. Corresponde a uma forma de predição, em que o valor a ser predito é um valor categórico. No caso em que o valor a ser predito é um valor numérico, trata-se de regressão (Bramer, 2007).

Han e Kamber (2006) descrevem a classificação como o processo de encontrar um modelo (ou função) que descreva e distinga classes de dados, com o objetivo de ser capaz de usar o modelo para prever a classe de exemplos cujo rótulo da classe é desconhecido. O modelo resulta da análise de um conjunto de dados de treino (isto é, os exemplos de dados cujos rótulos da categoria são conhecidos) (Han e Kamber, 2006).

Neste estudo será abordada a tarefa preditiva em que o valor a ser predito é um valor categórico, ou seja, a aplicação a considerar será a classificação, em particular, a classificação de documentos de textos. Este tópico será detalhado mais à frente (subcapítulo 2.4.).

2.2. *Text Mining*

As bases de dados textuais contêm normalmente frases longas ou parágrafos, como por exemplo, mensagens de alerta, relatórios de síntese, notas ou outros documentos. As bases de dados textuais podem ser não-estruturadas (por exemplo, páginas da *web*), semiestruturadas (por exemplo, e-mails) ou estruturadas (por exemplo, o catálogo da biblioteca) (Han e Kamber, 2006).

A metodologia de *text mining* surgiu pela primeira vez em meados dos anos 80, associada à possibilidade de extrair informação relevante de dados de texto. A rápida evolução da Internet e o aparecimento de mecanismos de busca fez com que *text mining* alcançasse uma maior importância nos anos 90. Nos últimos anos, a Internet contínua a revolucionar a comunicação e o acesso à informação. Milhões de pessoas usam a

Internet para procurar informações sobre diversas temáticas, tendo à disposição informação atualizada e melhorada ao longo do tempo. Estima-se que a percentagem de dados armazenados como texto possa chegar a 80 por cento [5]. A falta de ferramentas capazes de analisar este tipo de dados levou a que fosse realizada uma maior investigação nesta área nos últimos anos.

Feldman e Sanger (2007) definem *text mining* como um processo de conhecimento intensivo no qual o utilizador interage com uma coleção de documentos ao longo do tempo usando um conjunto de ferramentas de análise.

Analogamente a *data mining*, *text mining* refere-se ao processo de extração de conhecimento a partir de fontes de dados, através da identificação e exploração de padrões interessantes. A principal diferença recai na forma dos dados. No caso de *text mining*, são coleções de documentos com dados textuais não estruturados, enquanto, em *data mining* os dados são estruturados (Feldman e Sanger, 2007). Apesar da composição dos dados ser diferente, os métodos de aprendizagem são semelhantes. Nas duas metodologias, a aprendizagem é baseada em amostras de experiências passadas. Quando se está perante dados textuais, estes devem ser preparados e transformados numa representação numérica antes de qualquer método de aprendizagem poder ser aplicado (Weiss *et al*., 2010). Assim, uma etapa importante na metodologia *text mining* é a preparação textual. As operações de pré-processamento são responsáveis pela transformação de dados não estruturados num formato estruturado (Feldman e Sanger, 2007). A Figura 2 ilustra um exemplo de dados do tipo não estruturado.

> Em Março de 2004 abriu a loja de venda ao público e passou a disponibilizar todo o material informático, bem como todos os serviços técnicos associados.
> Em Agosto de 2005, com o aumento do negócio foi alterada a designação social para X JASF, Lda. pelos sócios José Ferreira e Célia Alves.
> Disponibilizamos todo o equipamento, bem como serviço técnico qualificado ao dimensionamento e execução de qualquer rede/ sistema de telecomunicações.

Figura 2: Exemplo de dados não estruturados

2.3. Pré-Processamento

O documento de texto inicial pode sofrer diversas alterações, através de diferentes técnicas de pré-processamento. As técnicas de processamento de dados, quando aplicadas antes da extração, podem melhorar substancialmente a qualidade geral dos padrões extraídos e/ou o tempo necessário para a extração efetiva (Han e Kamber, 2006).

De seguida são abordadas algumas tarefas de pré-processamento aplicadas neste estudo.

- **Transformar as letras maiúsculas em minúsculas**

As letras maiúsculas são convertidas em minúsculas para evitar que a mesma palavra, ao aparecer escrita de uma forma diferente, seja considerada como sendo uma palavra diferente. Por exemplo, se a palavra 'empresa' aparecer na coleção de documentos escrita das seguintes formas: 'empresa', 'Empresa' e 'EMPRESA', são contabilizadas 3 palavras diferentes. Com esta tarefa de pré-processamento as palavras passam a estar escritas da seguinte forma: 'empresa', 'empresa' e 'empresa'. Neste caso, é considerada uma só palavra que aparece 3 vezes.

- **Remover as *stopwords***

Stopwords são termos que por serem muito frequentes num documento quase não têm informação de maior relevância, ou seja, a sua entropia é muito baixa (Feinerer *et al.*, 2008). O principal objetivo da remoção de *stopwords* é eliminar termos com pouca informação.

A Tabela 2 apresenta a lista de *stopwords* portuguesas disponível no R, a qual foi obtida através do seguinte comando:

```
stopwords(language = "pt")
```

Tabela 2: Lista de *stopwords* portuguesas obtidas no R

a	com	é	este	for	houverem	meu	parte	sem	tenhamos	tivermos
à	como	ela	esteja	fora	houveremos	meus	pegar	ser	tenho	tivesse
acerca	comprido	elas	estejam	foram	houveria	minha	pela	será	tentar	tivessem
agora	conhecido	ele	estejamos	fôramos	houveriam	minhas	pelas	serão	tentaram	tivéssemos
algmas	corrente	eles	estes	forem	houveríamos	muito	pelo	serei	tente	todos
alguns	da	em	esteve	formos	houvermos	muitos	pelos	seremos	tentei	trabalhar
ali	das	enquanto	estive	fosse	houvesse	na	pessoas	seria	terá	trabalho
ambos	de	então	estivemos	fossem	houvessem	não	pode	seriam	terão	tu
antes	debaixo	entre	estiver	fôssemos	houvéssemos	nas	poderá	seríamos	terei	tua
ao	dela	era	estivera	fui	iniciar	nem	podia	seu	teremos	tuas
aos	delas	eram	estiveram	há	inicio	no	por	seus	teria	último
apontar	dele	éramos	estivéramos	haja	ir	nome	porque	só	teriam	um
aquela	deles	essa	estiverem	hajam	irá	nos	povo	somente	teríamos	uma
aquelas	dentro	essas	estivermos	hajamos	isso	nós	promeiro	somos	teu	umas
aquele	depois	esse	estivesse	hão	ista	nossa	qual	sou	teus	uns
aqueles	desde	esses	estivessem	havemos	iste	nossas	qualquer	sua	teve	usa
aqui	desligado	esta	estivéssemos	hei	isto	nosso	quando	suas	tinha	usar
aquilo	deve	está	estou	horas	já	nossos	que	tal	tinham	valor
as	devem	estado	eu	houve	lhe	novo	quê	também	tínhamos	veja
às	deverá	estamos	fará	houvemos	lhes	num	quem	te	tipo	ver
até	direita	estão	faz	houver	ligado	numa	quieto	tem	tive	verdade
atrás	diz	estar	fazer	houvera	maioria	o	saber	tém	tivemos	verdadeiro
bem	dizer	estará	fazia	houverá	maiorias	onde	são	têm	tiver	você
bom	do	estas	fez	houveram	mais	os	se	temos	tivera	vocês
cada	dois	estava	fim	houvéramos	mas	ou	seja	tempo	tiveram	vos
caminho	dos	estavam	foi	houverão	me	outro	sejam	tenha	tivéramos	
cima	e	estávamos	fomos	houverei	mesmo	para	sejamos	tenham	tiverem	

- **Remover a pontuação, números e espaços em branco**

O objetivo da remoção da pontuação, números e espaços nulos é eliminar termos com pouca informação para este estudo. A Tabela 3 apresenta uma lista com os sinais gráficos e números removidos após estas tarefas de pré-processamento.

Tabela 3: Lista com a pontuação e números removidos

.	**,**	**;**	**:**	**1**	**4**	**7**
?	**!**	**-**	**…**	**2**	**5**	**8**
"	**(**	**)**	**0**	**3**	**6**	**9**

- **Stemming**

Stemming consiste em reduzir palavras a uma forma comum. O processo consiste em apagar o sufixo da palavra para recuperar o seu radical (*stem*). É uma técnica

comum usada em *text mining* (Feinerer *et al.*, 2008). Weiss *et al.* (2010) referem que, para efeitos de classificação de documentos, a tarefa *stemming* pode beneficiar positivamente alguns casos. Um dos efeitos de *stemming* é a redução do número de atributos diferentes num corpo de texto e o aumento da frequência de ocorrência de alguns atributos individuais (Weiss *et al.*, 2010).

Feinerer *et al.* (2008) referem o algoritmo Porter (1997) como um dos melhores algoritmos de *stemming*. Trata-se de um algoritmo escrito para palavras inglesas.

No R, os *packages Rstem* e *Snowball* têm funções com o algoritmo de *stemming* de *Porter* implementado que permitem que o algoritmo possa ser usado em várias línguas, entre as quais a língua portuguesa.

O comando seguinte permite verificar para que línguas o algoritmo de *Porter* está adaptado no R (Feinerer *et al.*, 2008).

```
getStemLanguages()
 [1] "french"    "english"   "spanish"   "portuguese" "german"    "dutch"     "swedish"
"norwegian"  "danish"    "russian"   "finnish"
```

No entanto, como os algoritmos de *stemming* dependem das regras de formação de palavras para os quais foram escritos, existem alguns problemas da aplicação nas outras línguas (Honrado *et al.*, 2000).

- **Representação dos Documentos**

A coleção de documentos de texto tem que ser convertida para um formato estruturado para que os métodos de aprendizagem possam ser aplicados.

Uma representação comum é a representação como modelo de espaço vetorial. Neste modelo, um documento d_i do conjunto de documentos $D = \{d_1,...,d_n\}$, baseia-se num conjunto de palavras do dicionário $\{v_1,...,v_m\}$. O documento d_i pode ser representado por um vetor de dimensão *m*. A *i*-ésima componente do vetor é a frequência da *i*-ésima

palavra no dicionário, que apareceu no documento d_i. Assim, um conjunto de documentos que contém *n* documentos pode ser representado por uma matriz de documentos por termos (*Document-Term Matrix* – DTM) com *n* linhas e *m* colunas, onde *m* é o número de termos no dicionário. Muitos algoritmos de classificação de texto baseiam-se no modelo de espaço vetorial (Awad *et al.*, 2009). Este tipo de representação ignora a ordem das palavras, as combinações que delas ocorrem, a estruturação de parágrafo, pontuação e os significados das palavras. Um documento é considerado como sendo uma coleção de palavras simples que ocorrem pelo menos uma vez, ou seja, o documento é visto como um saco-de-palavras (*bag-of-words* – BOW) (Bramer, 2007).

Assim, a coleção de documentos pode ser transformada numa matriz, onde cada linha representa um documento e cada coluna representa um termo (palavra). A função DocumentTermMatrix() disponível no *package* 'tm' do R, faz a transformação descrita anteriormente (Feinerer *et al.*, 2008; Feinerer, 2011).

A DTM pode ser preenchida, por exemplo, por uns ou zeros, representando a presença ou não dos termos nos documentos, como pode ser observado na Figura 3.

	Terms						
Docs	ainda	antecipar	armazém	armazenagem	associação	augusta	banda
015.txt	0	0	0	1	1	0	0
016.txt	0	0	0	0	0	0	0
017.txt	0	0	0	0	0	0	0
018.txt	0	0	0	0	0	0	0
019.txt	1	0	0	0	0	1	1
020.txt	0	0	0	0	0	0	0
021.txt	1	0	0	0	0	0	0
022.txt	0	0	0	0	0	0	0
023.txt	0	1	0	0	0	0	0
024.txt	0	0	0	0	0	0	0
025.txt	0	0	1	0	0	0	0

Figura 3: Exemplo de uma DTM

Para conseguir melhorar a precisão da previsão, podem ser consideradas transformações adicionais a partir da representação da coleção de documentos (Weiss *et al.*, 2010).

O comando DocumentTermMatrix(docs_proc, control=list()) tem como parâmetro de entrada o control=list() que permite que seja inserida uma lista de filtros que se queiram aplicar (Feinerer *et al.*, 2008; Feinerer, 2011). Os filtros mais importantes referem-se a pesos e frequência de palavras. Assim, em vez de zeros ou uns como entradas nas células da matriz, pode ser utilizada a frequência real de ocorrência da palavra. Por exemplo, se uma palavra ocorre 10 vezes num documento, esta contagem seria inserida na célula (Weiss *et al*., 2010). Esta formulação está representada em (1).

- ✓ weighting=*weight*Tf – associa o peso da palavra i num documento j à sua frequência no documento.

$$w_{ij} = f_{ij} \tag{1}$$

Uma outra formulação, *tf-idf* (*term frequency - inverse document frequency*) considera a importância da palavra e é utilizada para calcular ponderações para as palavras. Em (2) está representada a formulação *tf-idf*.

- ✓ weighting=*weight*TfIdf – associa o peso de uma palavra i num documento j à multiplicação da sua frequência, no documento, pelo logaritmo da divisão do número total de documentos (N) pelo número de documentos que contém a palavra (n_i) (Witten e Frank, 2005).

$$w_{ij} = f_{ij} * \log\left(\frac{N}{n_i}\right) \tag{2}$$

Outros filtros podem ser acrescentados sem adicionar muita complexidade ao modelo e que podem reduzir significativamente o conjunto de dados.

- ✓ minWordLength - número mínimo de caracteres a considerar nas palavras.
- ✓ minDocFreq - número mínimo de documentos onde em as palavras aparecem.

- **Seleção de Características**

A representação BOW de um conjunto de documentos pode ser muito grande. A seleção de características é uma tarefa de pré-processamento que remove as palavras irrelevantes (Feldman e Sanger, 2007). O processo de seleção de características pode ser usado para remover termos dos documentos do conjunto de treino que são estatisticamente não correlacionados com os rótulos da categoria. Este processo irá reduzir o conjunto de termos a ser utilizado na classificação, melhorando assim a eficiência e precisão (Han e Kamber, 2006). Segundo Feldman e Sanger (2007), a maioria desses termos são irrelevantes para a tarefa de classificação e podem ser eliminados sem prejudicar o desempenho do classificador (Feldman e Sanger, 2007).

A fim de executar a seleção de características é necessário definir-se uma medida da importância de cada característica. Entre as diversas medidas existentes, a medida abordada neste estudo será o ganho de informação (*information gain*). O ganho de informação mede o número de bits de informação obtida para a predição de categorias, pelo conhecimento da presença ou ausência da característica *t* num documento (Feldman e Sanger, 2007). O ganho de informação do termo *t* com a classe c_k é obtido por (3).

$$IG(t) = -\sum_{k=1}^{C} p(c_k) \log_2 p(c_k) , \quad (3)$$

Onde $p(c_k)$ é a probabilidade de ocorrer a categoria c_k e *C* é o número de categorias. Apenas os atributos com os maiores valores de ganho de informação são mantidos para usar como *input* no algoritmo de classificação escolhido.

2.4. Classificação de Documentos

A classificação tem dois significados distintos. Uma abordagem é conhecida como aprendizagem não supervisionada, na qual se tem um conjunto de exemplos com o objetivo de estabelecer a existência de categorias, classes ou *clusters* nos dados; outra abordagem é a conhecida como aprendizagem supervisionada, segundo a qual se sabe o

número de categorias e o objetivo é estabelecer uma regra pela qual se pode classificar um novo exemplo numa das categorias existentes (Michie *et al.*, 1994).

Neste estudo será abordada apenas a classificação no caso da aprendizagem supervisionada, ou seja, o número de categorias e o seu significado é conhecido.

A tarefa de classificação de texto consiste na classificação de documentos de texto num número fixo de categorias pré-definidas com base no seu conteúdo. Sebastiani (2002) define a classificação de texto como a tarefa de atribuir um valor booleano para cada $(d_j, c_k) \in D \times C$, onde $D = \{d_1, \ldots, d_{|D|}\}$ é uma coleção de documentos e $C = \{c_1, \ldots, c_{|C|}\}$ é um conjunto de categorias pré-definidas. O valor 1, por exemplo, é atribuído a (d_j, c_k) com a decisão de atribuir o documento d_j à categoria c_k, enquanto o valor 0, por exemplo, é atribuído com a decisão contrária (Sebastiani, 2002).

Formalmente, a classificação de texto pode ser definida como a tarefa de aproximar uma função objetivo desconhecida $F: D \times C \rightarrow \{0, 1\}$ por meio de uma função $M: D \times C \rightarrow \{0, 1\}$ denominada de classificador ou modelo, tal que *M* produza resultados tão próximos quanto possível de *F* (Feldman e Sanger, 2007; Sebastiani, 2002).

Tan *et al*. (2006) definem a classificação como a tarefa de aprender uma função objetivo *f* que mapeia cada conjunto de atributos *x* para uma categoria pré-definida *y*. Segundo os mesmos autores, o *input* para a tarefa de classificação de documentos é uma coleção de registos, onde cada registo, também conhecido como uma instância ou exemplo, é caracterizado por uma *tupla* (x, y) em que *x* é o conjunto de atributos e *y* é um atributo especial, designado como etiqueta de classe (também conhecido como categoria ou atributo de destino) (Tan *et al.*, 2006).

A Figura seguinte mostra uma abordagem geral para a resolução de problemas de classificação.

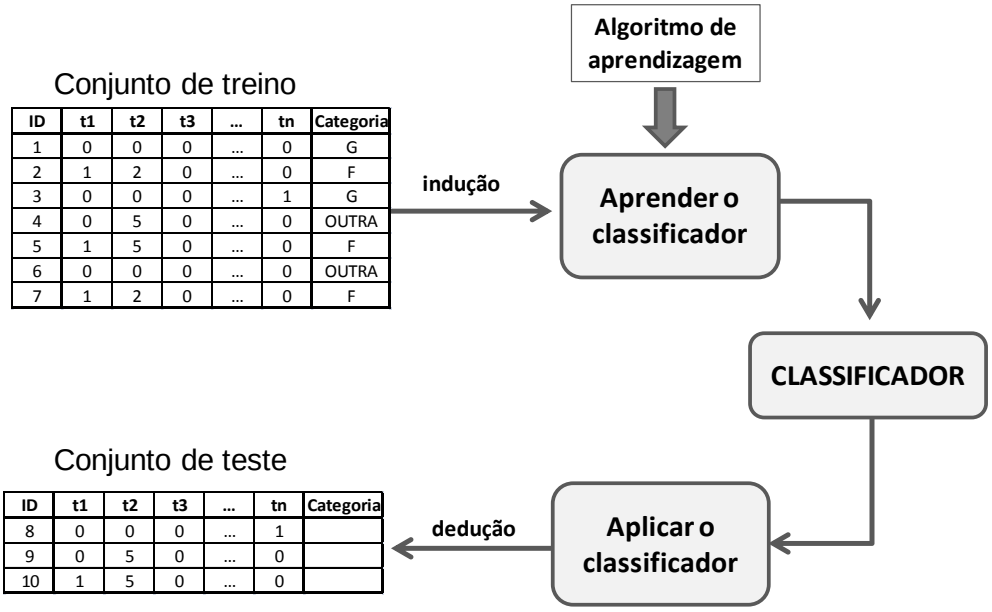


Figura 4: Abordagem geral para a resolução de problemas de classificação

Inicialmente deve ser fornecido um conjunto de treino composto de exemplos rotulados. O conjunto de treino é usado para construir um classificador, que é de seguida aplicado ao conjunto de teste, que consiste em exemplos com rótulos da categoria desconhecidos (Tan *et al.*, 2006).

Existem diferentes tipos de problemas de classificação de documentos, dependendo da aplicação pretendida. O caso em que exatamente uma categoria deve ser atribuída a cada documento é muitas vezes denominado uni-rótulo, enquanto o caso em que qualquer número de categorias pode ser atribuído ao mesmo documento é denominado o multi-rótulo. Um caso particular de classificação uni-rótulo é a classificação binária em que cada documento deve ser atribuído a apenas uma de duas categorias (Sebastiani, 2002). A classificação de documentos multi-classe consiste em classificar documentos em mais do que duas classes e a classificação multi-rótulo consiste em prever múltiplas classes para cada documento.

Os classificadores multi-classe também podem ser multi-rótulo, isto é, a resposta do classificador pode atribuir mais do que uma classe a um determinado exemplo (Silla e Freitas, 2011).

De acordo com a organização das categorias, o tipo mais usual de problema classificação é definido como classificação plana, ou seja, não existe uma relação hierárquica entre as categorias (Faceli *et al.*, 2011). Na Figura 5 pode observar-se um exemplo da estrutura da classificação plana. Mais à frente será abordada a classificação hierárquica bem como exemplos da estrutura referentes à organização das categorias.

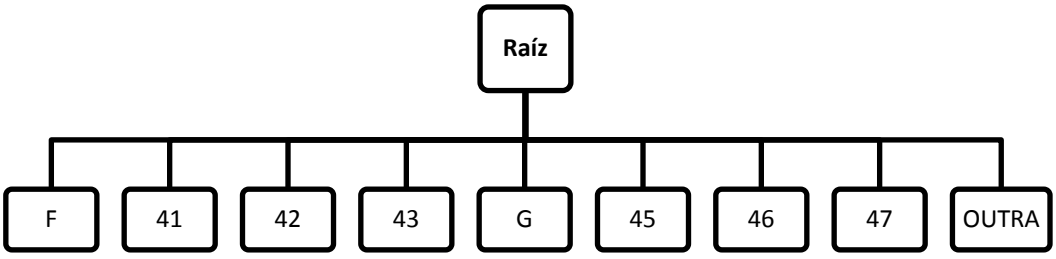


Figura 5: Exemplo de estrutura da classificação plana

2.4.1. Classificação Hierárquica

A classificação de documentos é denominada classificação hierárquica se as categorias estão organizadas hierarquicamente. Segundo Silla e Freitas (2011), a classificação hierárquica pode ser vista como um problema particular de classificação estruturada, na qual o *output* do algoritmo de classificação é definido através de uma taxonomia de categoria, enquanto a classificação estruturada refere-se a um problema de classificação, onde existe alguma estrutura (hierárquica ou não) entre as categorias (Silla e Freitas, 2011).

Nos problemas de classificação hierárquica, as categorias podem organizar-se em duas formas, em árvore ou num DAG (*Direct Acyclic Graph*). A diferença entre ambas recai no número de pais que um nó pode assumir, ou seja, na estrutura em árvore todos os nós têm um único pai, enquanto na estrutura DAG um nó pode ter mais do que um pai (Faceli *et al.*, 2011).

Neste estudo apenas será abordada a classificação hierárquica em árvore. Na Figura 6 é apresentada a estrutura de organização das categorias.

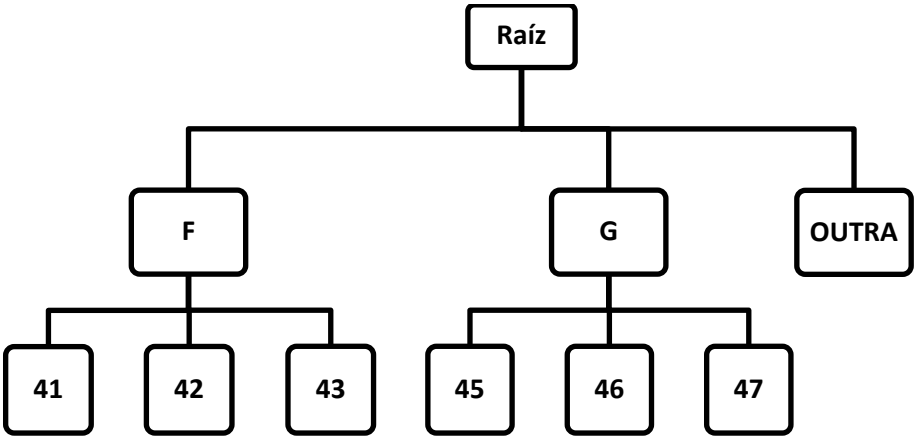


Figura 6: Estrutura hierárquica organizada em árvore

Segundo Silla e Freitas (2011), os classificadores binários e multi-classe não podem lidar diretamente com as categorias hierárquicas, consideram que estes tipos de classificadores não foram projetados para lidar com problemas de classificação hierárquica e referem-nos como algoritmos de classificação plana.

Os mesmos autores definem três tipos de algoritmos de classificação hierárquica local, são eles: classificação local por nó, classificação local por nó pai e classificação local por nível. Os três algoritmos diferem significativamente na sua fase de treino, no entanto, compartilham uma abordagem *top-down* (de cima para baixo) semelhante na fase de testes. Nesta abordagem, para cada novo exemplo no conjunto de teste, o sistema prevê primeiro a categoria do primeiro nível, em seguida, usa essa categoria prevista para estreitar as escolhas das categorias a serem previstas no segundo nível, e assim por diante, até que efetue a previsão mais específica. Uma desvantagem da abordagem *top-down* é que um erro num certo nível vai ser propagado a todos os nós descendentes (Silla e Freitas, 2011).

Neste estudo considera-se apenas a classificação local por nó pai e a classificação por nível, pois, segundo Silla e Freitas (2011), a classificação local por nó, consiste no treino de um classificador binário em cada nó da hierarquia de categorias, exceto no nó raiz, o que não se pretende neste estudo.

- **Classificação local por nó pai**

Se o classificador do primeiro nível atribui o exemplo à categoria 'G', o classificador do segundo nível só treina com os filhos do nó da categoria 'G', ou seja, com as categorias '45', '46' e '47'. A Figura 7 ilustra esta abordagem.

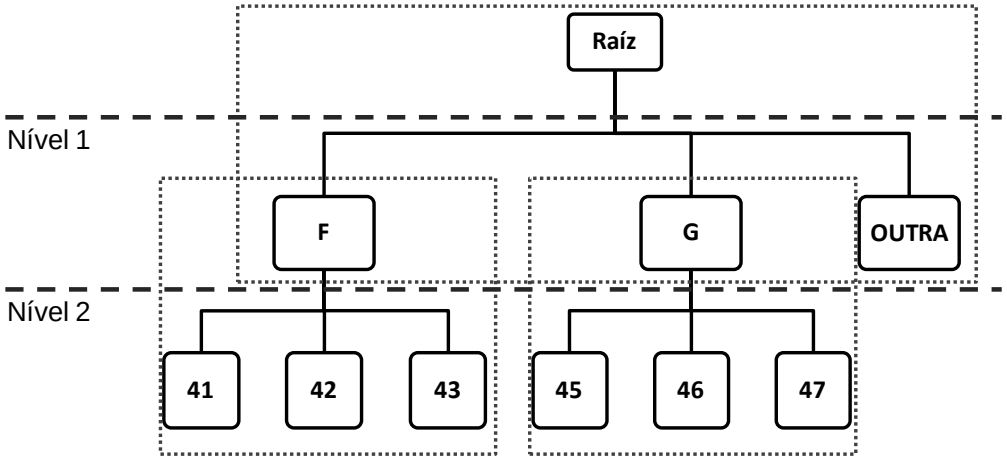


Figura 7: Classificação local por nó pai

- **Classificação local por nível**

O classificador local por nível consiste no treino de um classificador multi-classe para cada nível da hierarquia de classes. Segundo Silla e Freitas (2011), esta abordagem é referida na literatura com abordagem *top-down* (Silla e Freitas, 2011). Sun e Lim (2001) chamam a esta abordagem *top-down level-based*. Segundo os mesmos autores, na abordagem *top-down*, um ou mais classificadores são construídos em cada nível da árvore de categorias e cada classificador funciona como um classificador plano a esse nível (Sun e Lim, 2001). A Figura 8 ilustra esta abordagem. Neste tipo de classificação e considerando o esquema da Figura 8, dois classificadores serão treinados, um classificador para cada nível. No primeiro nível o classificador será treinado para prever 3 categorias, enquanto no segundo nível será treinado para prever 6 categorias.

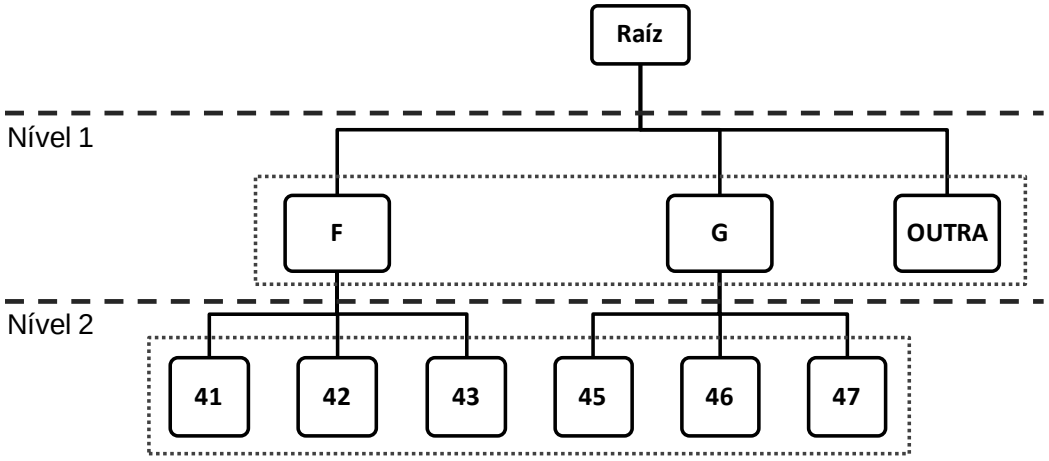


Figura 8: Classificação local por nível

2.5. Algoritmos de Classificação

A técnica de classificação consiste na construção de classificadores a partir de um conjunto de dados de entrada. O classificador emprega um algoritmo de aprendizagem para identificar um modelo (classificador) que melhor ajusta a relação entre o conjunto de atributos e a categoria rotulada dos dados de entrada, com o objetivo de prever corretamente os rótulos da categoria de novos exemplos. Portanto, o objetivo principal do algoritmo de aprendizagem é construir classificadores capazes de prever com precisão os rótulos de categoria de exemplos anteriormente desconhecidos (Tan *et al.*, 2006).

Os algoritmos de aprendizagem automática (*machine learning*) abordados neste estudo e que serão descritos de seguida são as árvores de decisão, os *k*-vizinhos mais próximo, redes neuronais, *Naive Bayes* e *support vector machines*.

2.5.1. Árvores de Decisão

O algoritmo de árvores de decisão é um método baseado em procura, que produz uma estrutura de árvore com base nos exemplos do conjunto de treino.

Na resolução de problemas de aprendizagem em conjuntos de dados independentes é utilizada uma forma de representação gráfica – a Árvore de Decisão, trata-se de um gráfico de fluxo em forma de árvore, composto por vários nós correspondentes aos diferentes atributos. A árvore é composta pelos seguintes elementos: nó raiz, nós internos, ramos e nós folha. O nó raiz é o nó que inicia a árvore, os nós internos representam um teste a um atributo, os ramos representam o resultado do teste e os nós folha são portadores da informação da categoria. Pela sua simplicidade e facilidade de utilização, as árvores de decisão são bastante usadas, também por fornecerem informação que é analisada intuitivamente e é facilmente assimilável. No entanto, estas 'facilidades' não indicam que este tipo de classificador fornece resultados de fraca qualidade, muito pelo contrário, as árvores de decisão podem ser bastante precisas e são aplicáveis nas mais variadas áreas de atividade (Han e Kamber, 2006).

A árvore de decisão classifica um documento, iniciando na raiz da árvore e move-se, sucessivamente, para baixo, através dos ramos cujas condições são satisfeitas pelo documento até que um nó de folha é atingido. O documento é então atribuído à categoria que rotula o nó folha (Feldman e Sanger, 2007).

2.5.2. *k*-NN (*k*-Nearest Neighbor)

O algoritmo *k*-NN ou algoritmo dos vizinhos mais próximo é um método baseado em distâncias sendo a distância euclidiana a mais usual (ver (4)).

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4}$$

Em que *n* é o número de termos e x_i e y_i são os *i*-ésimos termos *x* e *y*. Esta medida é usada para calcular a distância entre dois documentos.

A classificação de um novo exemplo é feita considerando os *k* exemplos mais próximos a esse exemplo, ou seja, o exemplo de teste é classificado com base nos exemplos do conjunto de treino mais próximos a ele. Em problemas de classificação, como é o caso deste estudo, considera-se a categoria mais frequente entre os *k*-vizinhos considerados.

O algoritmo dos vizinhos mais próximos é considerado preguiçoso (*lazy learner*), porque não aprende um modelo compacto para os dados. É o mais simples de todos os algoritmos de aprendizagem automática (Faceli *et al*., 2011).
Feldman e Sanger (2007) consideram o *k*-NN um dos classificadores com melhor desempenho em classificação de texto. É robusto no sentido de não exigir que as categorias sejam linearmente separadas. A sua única desvantagem é relativa ao elevado custo computacional de classificação, isto é, para cada documento de teste deve ser calculada a distância desse documento a todos os documentos de treino (Feldman e Sanger, 2007).

2.5.3. Redes Neuronais

O algoritmo rede neuronal é um método baseado em otimização. Numa rede neural é dado um conjunto de entradas que é utilizado para prever uma ou mais saídas (Bramer, 2007). O termo rede neuronal aplicado à classificação de dados teve a sua origem pela analogia que existe entre a rede cerebral, composta por neurónios interligados, e uma rede de dados com elementos de processamento, também interligados, em que ambos respondem a estímulos (no caso do cérebro) ou a dados introduzidos (no caso da classificação de dados).

Uma rede neuronal, quando usado para a classificação, é tipicamente uma coleção de unidades de processamento (neurónios) com conexões ponderadas entre as unidades (Han e Kamber, 2006).

Segundo Russell (1996), uma das características mais importantes de uma rede neuronal é a sua facilidade de adaptação a novos ambientes. A aprendizagem é essencial para a maioria das arquiteturas de redes neuronais e assim, a escolha de um algoritmo de aprendizagem é fulcral no desenvolvimento da rede. A aprendizagem implica que uma unidade de processamento é capaz de alterar o comportamento das suas entradas/saídas como resultado de alterações no ambiente.

Mitchell (1997) refere-se às redes neuronais como métodos de aprendizagem que fornecem uma abordagem robusta na aproximação das funções de valor real, de valor discreto e de valor vetorial. O mesmo autor acrescenta que a rede neuronal é, dos métodos conhecidos, um dos mais eficazes, de que é exemplo o algoritmo de retropropagação (*backpropagation*), que aprende os pesos de uma rede multicamada, dada uma rede com conjunto fixo de unidades e interligações.

2.5.4. *Naive Bayes*

O algoritmo *Naive Bayes* é um método de aprendizagem probabilística baseado no teorema de *Bayes*, apresentado em (5).

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \ , \tag{5}$$

onde A e B são dois acontecimentos e $P(A \mid B)$ é a probabilidade de A dado B.

Mihaescu (2011) considera este algoritmo fiável, com a vantagem de necessitar de uma pequena quantidade de dados de treino para estimar os parâmetros (Mihaescu, 2011). Segundo Han e Kamber (2006) a classificação *bayesiana* é uma das várias técnicas que podem ser utilizadas para uma classificação de documentos eficaz (Han e Kamber, 2006).

Segundo Mihaescu (2011), a classificação *bayesiana* representa um método de aprendizagem supervisionada, bem como um método estatístico de classificação. Assume um modelo probabilístico subjacente e permite determinar a incerteza associada ao modelo, em princípio, pela determinação das probabilidades dos resultados e pode resolver problemas de diagnóstico e previsão. A classificação *bayesiana* aporta algoritmos de aprendizagem que poderão ser combinados com dados de conhecimentos e observações anteriores, contribuindo para o seu entendimento e avaliação. Os algoritmos calculam a probabilidade explícita da hipótese, para além de serem robustos ao ruído nos dados de entrada.

Estes classificadores são particularmente utilizados quando a dimensionalidade dos dados de entrada (*input*) é elevada. A estimação de parâmetros em modelos *Naive Bayes* utiliza o método de máxima verosimilhança (Mihaescu, 2011).

Apesar das assunções muito simplistas, estes modelos funcionam bastante bem em situações extremamente complexas do mundo real.

2.5.5. SVM (*Support Vector Machines*)

O algoritmo SVM é um método baseado em otimização. De acordo com Press *et al.* (2007), o classificador SVM foi, primeiramente, descrito por Vapnik *et al.* (1992), e estabeleceu-se rapidamente, por si só, como uma ferramenta poderosa no estudo e resolução de problemas de classificação, que até então eram resolvidos por redes neuronais e por outros métodos, todos de complexidade elevada. Feldman e Sanger (2007) consideram o algoritmo SVM como muito rápido e eficaz para problemas de classificação de texto.

O SVM é uma ferramenta de fácil implementação, tem uma interface bastante intuitiva e os resultados obtidos são mais 'cristalinos', ao contrário da opacidade dos obtidos por redes neuronais (Press *et al.*, 2007).

Segundo Feldman e Sanger (2007), o classificador SVM tem uma vantagem importante para o problema *overfitting* (sobreajustamento), o que lhe permite ter um bom desempenho, independentemente da dimensionalidade do espaço de características. Além disso, este algoritmo não precisa de ajuste de parâmetros, porque são ajustados experimentalmente, por defeito, para proporcionar um melhor desempenho (Feldman e Sanger, 2007).

SVM Linear

Um classificador SVM linear pode ser visto, em termos geométricos, como um hiperplano do espaço de atributos que separa os pontos que representam os exemplos

positivos dos pontos que representam os exemplos negativos. Os hiperplanos SVM são determinados por um subconjunto relativamente pequeno dos exemplos de treino, denominados de vetores de suporte. O resto dos dados de treino não tem qualquer influência sobre o classificador treinado. A este respeito, o algoritmo SVM parece ser único entre os diferentes algoritmos de classificação (Feldman e Sanger, 2007). Na Figura 9 é possível visualizar um SVM linear.

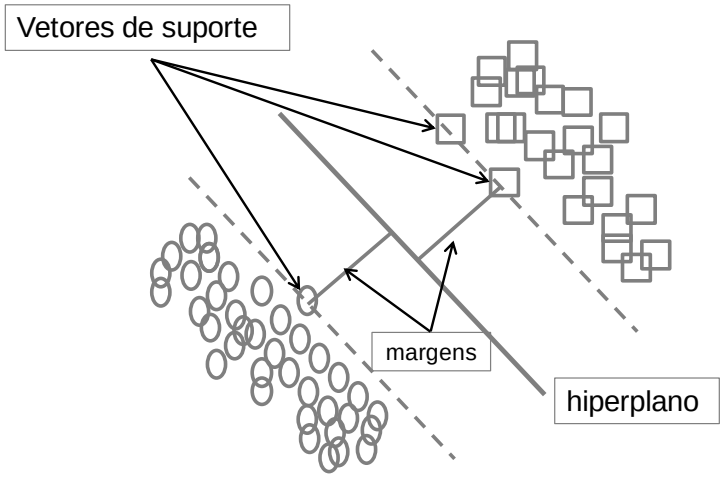


Figura 9: SVM linear

2.6. Métodos de Avaliação

A avaliação do desempenho de um classificador baseia-se nas contagens de exemplos de teste corretamente e incorretamente previstos pelo classificador (Tan *et al.*, 2006).

A matriz de confusão facilita a visualização do número de classificações corretas e do número de classificações preditas para cada classe, de um determinado conjunto de exemplos, segundo o classificador em análise. Torna-se uma ferramenta útil para analisar a qualidade do classificador no reconhecimento de exemplos de diferentes categorias (Han e Kamber, 2006).

Quando um conjunto de dados tem apenas duas categorias, é muitas vezes considerada uma como 'positiva' e a outra como 'negativa'. As entradas da matriz de confusão são referidas como verdadeiros positivos (*true positives* - TP), falsos positivos (*false*

positives - FP), falsos negativos (*false negatives* - FN) e verdadeiros negativos (*true negatives* - TN) (Bramer, 2007).

A tabela seguinte apresenta a matriz de confusão para um problema de classificação binária.

Tabela 4: Matriz de confusão para um problema de classificação de 2 categorias

Categoria	Prevista C^+	Prevista C^-
Verdadeira C^+	TP	FN
Verdadeira C^-	FP	TN

TP refere-se ao número de exemplos da categoria 'positiva' corretamente previstos como categoria 'positiva'; FN representa o número de exemplos da categoria 'positiva' incorretamente previstos como categoria 'negativa'. FP refere-se ao número de exemplos da categoria 'negativa' incorretamente previstos como categoria 'positiva' e TN representa o número de exemplos da categoria 'negativa' corretamente previstos como categoria 'negativa'.

Com base nas entradas da matriz de confusão, o número total de previsões corretas feitas pelo classificador é TP+TN e o total número de previsões incorretas é FP+FN.

O uso da matriz confusão não é limitado a problemas de classificação com duas categorias. Nos problemas de classificação multi-classe, o resultado de um conjunto de teste é muitas vezes apresentado como uma matriz de confusão bidimensional com uma linha e coluna para cada categoria. Cada elemento da matriz mostra o número de exemplos de teste para a qual a categoria verdadeira corresponde à linha e a categoria é predita corresponde à coluna (Witten e Frank, 2005).

A performance de um classificador pode ser expressa em termos da sua taxa de erro.

- Taxa de erro:

$$\text{Taxa de Erro} = \frac{\text{número de erros}}{\text{número de documentos}} \quad (6)$$

- Precisão (*precision*): A precisão para a categoria *c* é obtida através da expressão (7) e mede a percentagem de atribuições corretas entre todos os documentos atribuídos a *c*.

$$\text{Precisão}_c = \frac{TP_c}{TP_c + FP_c} \tag{7}$$

- Sensibilidade (*recall*): A medida *recall* para a categoria *c* é ilustrada em (8) e indica a percentagem de atribuições corretas em *c* entre todos os documentos que deviam ser atribuídos a *c*.

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c} \tag{8}$$

Faceli *et al*. (2011) consideram a precisão uma medida de exatidão do classificador e a *recall* uma medida da sua completude. A análise destas duas medidas em separado normalmente não é discutida. No entanto, a média harmónica ponderada das duas medidas dá origem à medida-F_β ilustrada em (9).

- Medida-F_β:

$$F_{\beta\,c} = \frac{(\beta+1) * \text{Precisão}_c * \text{Recall}_c}{\beta * \text{Precisão}_c + \text{Recall}_c} \tag{9}$$

Normalmente considera-se β=1, ou seja, atribui-se a mesma importância à precisão e à *recall*, ficando a fórmula representada em (10).

$$F_{1c} = \frac{2 * \text{Precisão}_c * \text{Recall}_c}{\text{Precisão}_c + \text{Recall}_c} \tag{10}$$

Existem dois métodos para calcular o desempenho de um sistema de classificação de texto com base na precisão e *recall*. Micro-média (*Micro-average*) e Macro-média *macro-average* (Sun e Lim, 2001).

- *Micro-average*: dá igual importância a cada documento e é calculado através da construção de uma tabela de contingência global e depois calcula-se a precisão (11) e a *recall* (12) considerando essa tabela. Esta medida é obtida através da expressão (13).

$$P_{micro} = \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C} TP_c + FP_c} \tag{11}$$

$$R_{micro} = \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C} TP_c + FN_c} \tag{12}$$

$$microF1 = 2 * \frac{P_{micro} * R_{micro}}{P_{micro} + R_{micro}} \tag{13}$$

- *Macro-average*: dá igual importância a cada categoria e é obtida pelo cálculo da precisão (14) e *recall* (15) para cada categoria e, em seguida, considerando a média destes. Esta medida está ilustrada em (16).

$$P_{macro} = \frac{1}{|C|} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c} \tag{14}$$

$$R_{macro} = \frac{1}{|C|} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FN_c} \tag{15}$$

$$macroF1 = 2 * \frac{P_{macro} * R_{macro}}{P_{macro} + R_{macro}} \tag{16}$$

2.7. Análise de Similaridade

Segundo Weiss *et al*. (2010), a medida de similaridade cosseno, apenas considera palavras positivas partilhadas para comparar os documentos, mas a frequência de

ocorrência de palavras também é valorizada. Colocar estes temas em conjunto, temos a medida distância cosseno apresentada em (17). O peso de uma palavra num documento *w*(*i*) é calculado pela fórmula *tf-idf* apresentada em (2). Note-se que *w*(*i*) é igual a zero, quando uma palavra não aparece num documento, de modo que apenas palavras partilhadas entre os dois documentos comparados são de interesse (Weiss *et al.*, 2010).

$$sim(\overrightarrow{d_1}, \overrightarrow{d_2}) = \cos(\overrightarrow{d_1}\overrightarrow{d_2}) = \frac{\overrightarrow{d_1} . \overrightarrow{d_2}}{\left|\overrightarrow{d_1}\right| . \left|\overrightarrow{d_2}\right|} = \frac{\sum_i w_{i,1} . w_{i,2}}{\sqrt{\sum_i w_{i,1}^2} \cdot \sqrt{\sum_i w_{i,2}^2}} \quad (17)$$

Onde, $w_{i,j}$ é o peso do termo i no documento j.

A medida de similaridade cosseno retorna valores entre 0 e 1, quanto mais próximo de 1 for o valor, mais similares são os dois documentos.

CAPÍTULO 3

Classificação Hierárquica: Caso de Estudo

A classificação hierárquica de documentos consiste, como foi visto na secção anterior, na atribuição de um número de classes estruturadas hierarquicamente, a documentos de texto. Neste capítulo faz-se uma aplicação das técnicas anteriormente estudadas, a documentos de texto com o descritivo da atividade económica das empresas obtido a partir da página da Internet (*webpage*) de cada uma delas.

Neste capítulo 3 começa-se por apresentar a Classificação Portuguesa das Atividades Económicas, Revisão 3 (CAE-Rev.3), instrumento de referência para a classificação das empresas em Portugal. Esta nomenclatura é usada para definir as categorias usadas no processo de classificação. Em seguida apresenta-se o processo de recolha de documentos sobre um dado conjunto de empresas e a coleção de documentos recolhida. Posteriormente, é efetuado o pré-processamento dos documentos e a separação dos mesmos no conjunto de treino e conjunto de teste. Como o objetivo do conjunto de treino é gerar classificadores que permitam processar os documentos e prever as respetivas categorias, neste capítulo será abordada a seleção de características do conjunto de treino com a finalidade de remover termos estatisticamente não correlacionados com os rótulos da categoria. Finalmente será preparado o conjunto de documentos necessários à análise de similaridade, bem como a descrição de tarefas intermédias.

3.1. Abordagem ao Problema

A diversidade de algoritmos de classificação e os diferentes tipos de problemas de classificação hierárquica, exigem que seja descrita de uma forma mais precisa o tipo de problema de classificação hierárquica em estudo.

O problema de classificação hierárquica proposto refere-se a um caso de classificação multi-classe uni-rótulo, ou seja, o número de categorias a usar no problema de classificação é superior a duas e cada documento pode apenas ser classificado numa única categoria. Trata-se de um problema de aprendizagem supervisionada, pois as categorias são pré-definidas. As categorias estão organizadas hierarquicamente numa estrutura em árvore.

O objetivo principal deste trabalho consiste em classificar, segundo uma estrutura hierárquica em árvore (ver Figura 6), uma coleção de documentos. A classificação hierárquica a ser usada vai seguir duas metodologias diferentes. A primeira metodologia considera a classificação local por nó pai (ver Figura 7) e a segunda considera a classificação local por nível (ver Figura 8). Para além do objetivo principal, são apontados dois objetivos secundários. Um corresponde a analisar os efeitos de variar o método (documentos sem *stemming* e documentos com *stemming*), enquanto o segundo consiste em classificar os documentos segundo a análise de similaridade entre os documentos com a descrição das empresas e os documentos com a descrição das categorias.

Os documentos com a descrição das empresas correspondem à descrição efetuada pela empresa na página da Internet sobre a sua atividade económica e os documentos da descrição das categorias correspondem à descrição disponível na CAE Rev.3 para cada uma das categorias usadas no problema de classificação.

3.2. Classificação da Atividade Económica

A atividade económica das empresas está classificada segundo a CAE-Rev.3 produzida em 2007 pelo Instituto Nacional de Estatística (INE), com a colaboração de algumas entidades e empresas (INE, 2007).

A CAE-Rev.3 está em conformidade com a Nomenclatura Estatística das Atividades Económicas na Comunidade Europeia (NACE-Rev.2) e está disponível no *site* (http://metaweb.ine.pt/sine). A classificação da atividade económica resulta da

combinação de fatores produtivos com vista à produção de bens e serviços. As empresas praticam, na maioria das vezes, mais do que uma atividade económica. Nestes casos, a empresa é classificada segundo a sua atividade principal, que corresponde à atividade de maior importância no conjunto das atividades praticadas pela empresa.

Segundo o INE, a classificação de atividades económicas tem grande importância na vida das empresas e pode influenciar questões como o imposto sobre o valor acrescentado (IVA) a aplicar aos bens e/ou serviços de uma empresa (INE, 2007).

A estrutura da CAE-Rev. 3 é esquematizada na Figura 10.

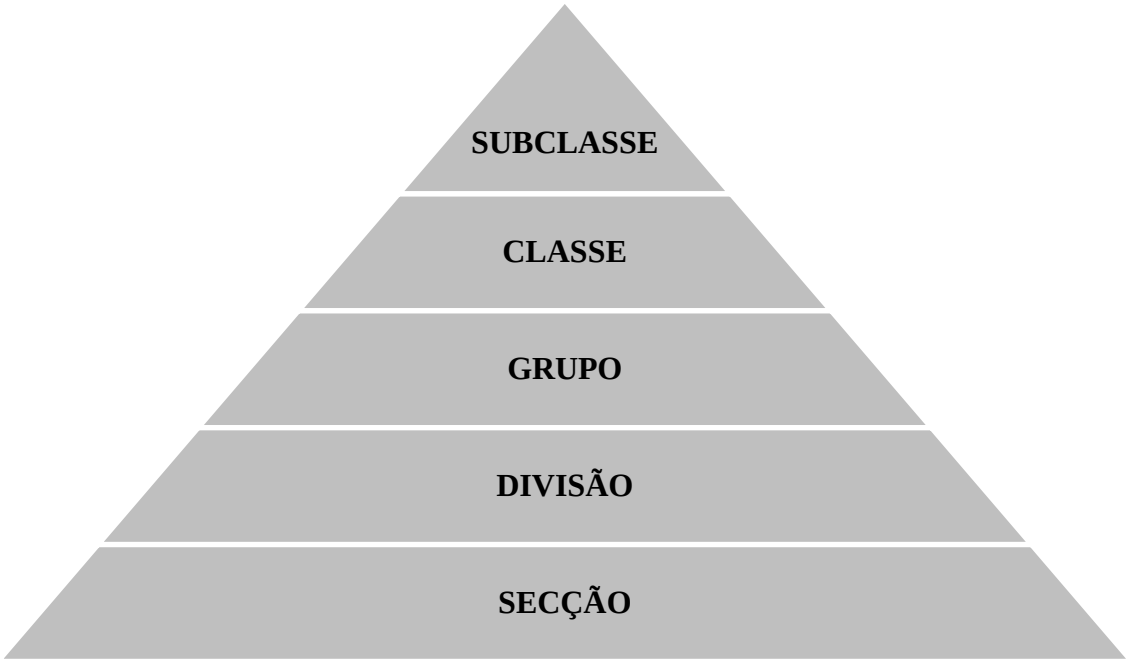


Figura 10: Estrutura da CAE-Rev.3

As Secções representam o primeiro nível e são identificadas através de um código alfabético; as Divisões representam o segundo nível e são identificadas através de um código de dois dígitos; os Grupos representam o terceiro nível e são identificados através de um código de três dígitos; as Classes representam o quarto nível e são identificadas através de um código de quatro dígitos; finalmente, as Subclasses representam o quinto nível e são identificadas através de um código de cinco dígitos.

O código CAE-Rev. 3 permite, ao nível da análise estatística, classificar e agrupar as empresas, segundo a atividade económica; comparar estatísticas a nível nacional e mundial. Ao nível da atividade económica, o código CAE-Rev.3 permite registar as empresas no ato da sua formação; promover o licenciamento das atividades económicas; apoiar as políticas do Governo de incentivos às atividades económicas [INE, 2007].

A atividade económica das empresas pode ser classificada de acordo com 21 Secções, divididas em 88 Divisões, que por sua vez representam 272 Grupos, estes são divididos em 615 Classes e finalmente, a um nível mais desagregado, as empresas podem ser classificadas em 848 Subclasses. A título de exemplo, pode observar-se a Figura 11 que esquematiza a estrutura hierárquica da Secção F. Nesse esquema, apenas a Divisão '41' está desagregada ao nível da subclasse, as restantes estão até ao nível de Grupo (associadas a esses Grupos estão 20 Classes que por sua vez têm associadas 22 Subclasses).

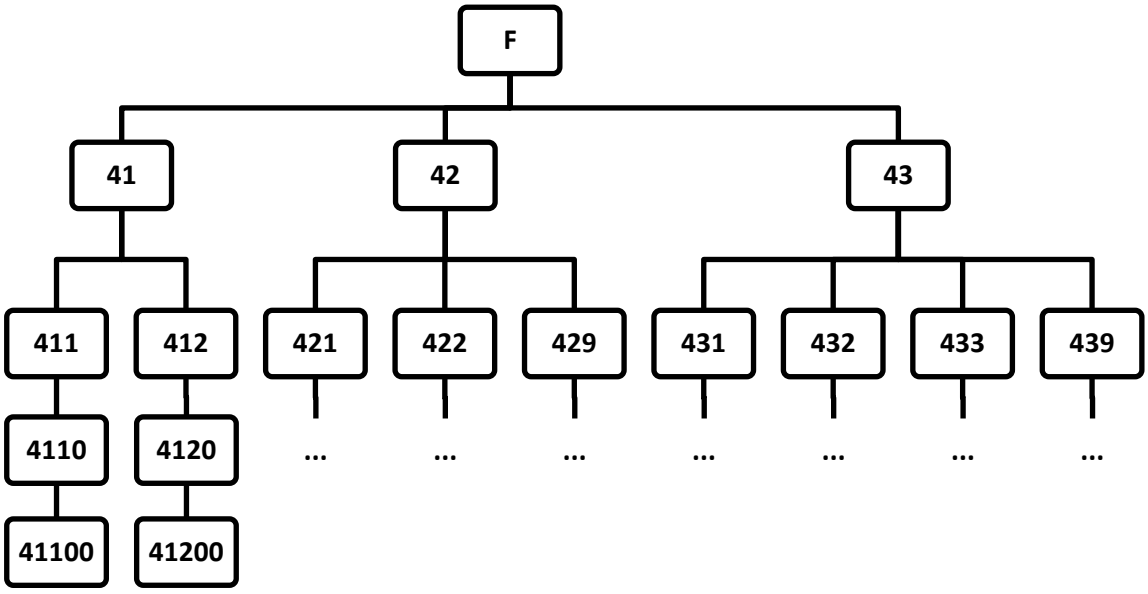


Figura 11: Hierarquia parcial da Secção 'F' até ao nível da Subclasse

As Secções podem ser divididas em Grupos, os Grupos em Divisões, as Divisões em Classes e as Classes em Subclasses. Trata-se de um problema de classificação em que as categorias podem ser divididas em subcategorias.

O elevado número de categorias disponíveis para classificação de documentos, levou a que fosse aplicada uma restrição do número das mesmas. Neste estudo optou-se por considerar a estrutura hierárquica observada na Figura 6, ou seja, são considerados apenas dois níveis, o primeiro nível contém as categorias 'F', 'G', e 'OUTRA' e o segundo nível contém as categorias '41', '42', '42', '45', '46', '47'. Esta opção surgiu do número de documentos de texto recolhidos, ou seja, no momento em que se optou por esta restrição, os documentos de texto pertencentes às Secções F e G estavam representados em maior número face às restantes Secções.

Detalhadamente, a categoria 'F' corresponde à Secção F que representa a construção. As empresas classificadas nesta categoria podem ainda ser classificadas nas Divisões 41 – promoção imobiliária (desenvolvimento de projetos de edifícios), construção de edifícios; 42 – engenharia civil e 43 – atividades especializadas de construção. A categoria 'G' corresponde à Secção G que representa o comércio por grosso e a retalho; reparação de veículos automóveis e motociclos. Ao nível mais desagregado, as empresas desta Secção podem ser classificadas nas Divisões 45 – comércio, manutenção e reparação, de veículos automóveis e motociclos; 46 – comércio por grosso (inclui agentes), exceto de veículos automóveis e motociclos; e 47 – comércio a retalho, exceto de veículos automóveis e motociclos. A categoria 'OUTRA' corresponde a algumas das restantes Secções (B - indústrias extrativas; C – indústrias transformadoras; D – eletricidade, gás, vapor, água quente e fria e ar frio; E – captação, tratamento e distribuição de água, saneamento, gestão de resíduos e despoluição; H – transportes e armazenagem; J – atividades de informação e de comunicação; K – atividades financeiras e de seguros; L – atividades imobiliárias, M – atividades de consultoria, científicas, técnicas e similares; N – atividades administrativas e dos serviços de apoio; P – Educação; Q – atividades de saúde humana e apoio social).

A relação das Secções com as Divisões pode ser vista no Anexo 1.

3.3. Recolha de Dados

Uma das tarefas mais demoradas deste trabalho foi a recolha de dados sobre a atividade económica das empresas. A não existência de uma base de dados com a informação do descritivo da atividade económica, disponibilizada pelas empresas na Internet, levou a que fosse necessário efetuar essa recolha previamente.

O pretendido era aceder ao *website* de cada uma das empresas e extrair o descritivo da sua atividade, para um ficheiro em formato texto, de forma automática. No entanto, devido ao elevado número de páginas *web* e diferentes formas de criação das páginas por parte das empresas, a automatização da recolha não foi possível.

O endereço de URL (*Uniform Resource Locator*) das empresas foi obtido de um diretório de empresas portuguesas disponível *online* em http://directorio.informadb.pt/, pertencente à Informa D&B [1]. Deste diretório, para além do endereço de URL da empresa também foi retirado o código CAE-Rev.3, referente à atividade principal, de cada empresa. Esta informação dá origem à variável 'categoria', pois a aprendizagem supervisionada pressupõe que o número de categorias e o seu significado seja conhecido. A Figura 12 mostra um exemplo de listagem de empresas disponíveis nesse diretório.

Lista de Empresas

Empresa	Concelho	Freguesia	Website
VANIBRU - COMERCIO DE PRODUTOS ALIMENTARES, LDA	Braga	Arcos (Braga)	www.vanibru.pt
AGROSPORT - PRODUTOS, EQUIPAMENTOS E TÉCNICA AGRÁRIA, LDA	Cartaxo	Cartaxo	www.agrosport.pt
OLIVEIRA & IRMÃO, S.A.	Aveiro	Requeixo	www.oliveirairmao.com
ROBALO - UTILIDADES DOMÉSTICAS E HOTELEIRAS, S.A.	Vila Franca De Xira	Vila Franca De Xira	
TEXTEIS ALMEIDA COIMBRA, LDA	Santo Tirso	Santo Tirso	www.tacmalhas.com
PLANOTÉCNICA - GABINETE DE ESTUDOS DE INSTALAÇÕES ELÉCTRICAS, LDA	Vila Franca De Xira	Póvoa De Santa Iria	
MANPOWER PORTUGUESA - SERVIÇOS DE RECURSOS HUMANOS (EMPRESA DE TRABALHO TEMPORÁRIO), S.A.	Lisboa	São Jorge De Arroios	www.manpower.pt
ÂNGELO COIMBRA & CA., LDA	Maia	Moreira (Maia)	www.angelocoimbra.pt
PLASTIDOM - PLÁSTICOS INDUSTRIAIS E DOMÉSTICOS, S.A.	Leiria	Marrazes	www.plastidom.pt
MALHAS EICAL - EMPRESA INDUSTRIAL DO CÁVADO, LDA	Barcelos	Mariz	www.eical.com
CORKART - INDÚSTRIA DE CORTIÇAS, S.A.	Vendas Novas	Vendas Novas	www.corkart.pai.pt

Figura 12: *Site* da Informa D&B com exemplo de listagem das empresas

A imagem seguinte mostra a informação disponibilizada no *site* da Informa D&B, sobre as empresas. Esta informação é obtida ao clicar no nome da empresa desejada, das observadas na listagem.

Figura 13: Identificação do código CAE-Rev.3 e do endereço de URL de uma empresa *no site* da Informa D&B

Posteriormente, acedido o *website* de cada empresa, foi extraída a informação pretendida para um ficheiro *.txt.

Figura 14: Exemplo do descritivo da atividade económica de uma empresa extraído da *webpage*

No final da recolha obteve-se uma coleção com 800 documentos de texto, em que cada documento corresponde ao descritivo, disponível na Internet, da atividade económica de cada uma das empresas consideradas. Essa informação foi retirada da respetiva página na Internet dos campos "quem somos" ou "o que fazemos" ou "empresa", na maior parte das vezes. No caso da empresa tida como exemplo na Figura 14, a informação foi extraída do campo "Empresa".

De seguida é apresentado um esquema ilustrativo do processo de recolha de documentos descrito anteriormente.

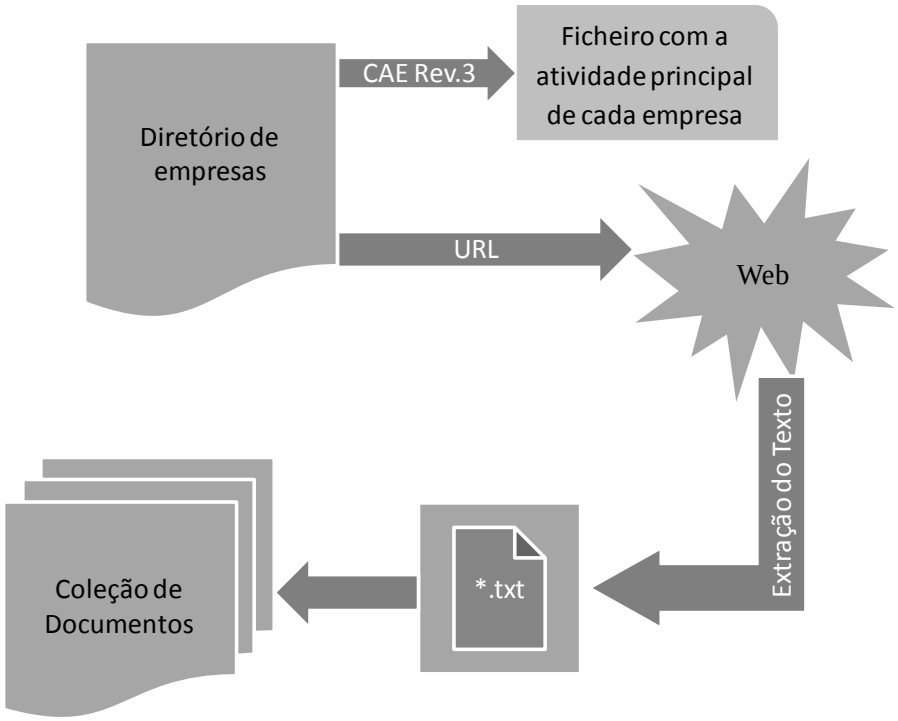


Figura 15: Esquema ilustrativo da recolha de documentos

3.4. Coleção de Documentos – Criação do Corpus

A coleção de documentos de texto que serve de base a este estudo contém 800 documentos de texto sobre a atividade económica das empresas. Os documentos estão escritos em português.

As tabelas seguintes apresentam a distribuição dos documentos considerados neste estudo por categorias do primeiro e segundo níveis, ou seja, por Secções e Divisões da CAE-Rev.3.

Tabela 5: Distribuição dos documentos por Categoria do 1º Nível

Categoria	Total de Documentos
F	256
G	251
OUTRA	293

Como referido anteriormente, documentos de diversas Secções foram agrupados numa categoria criada para este estudo e denominada por 'OUTRA'. Na Figura 16 é possível observar as Secções pertencentes à categoria 'OUTRA'.

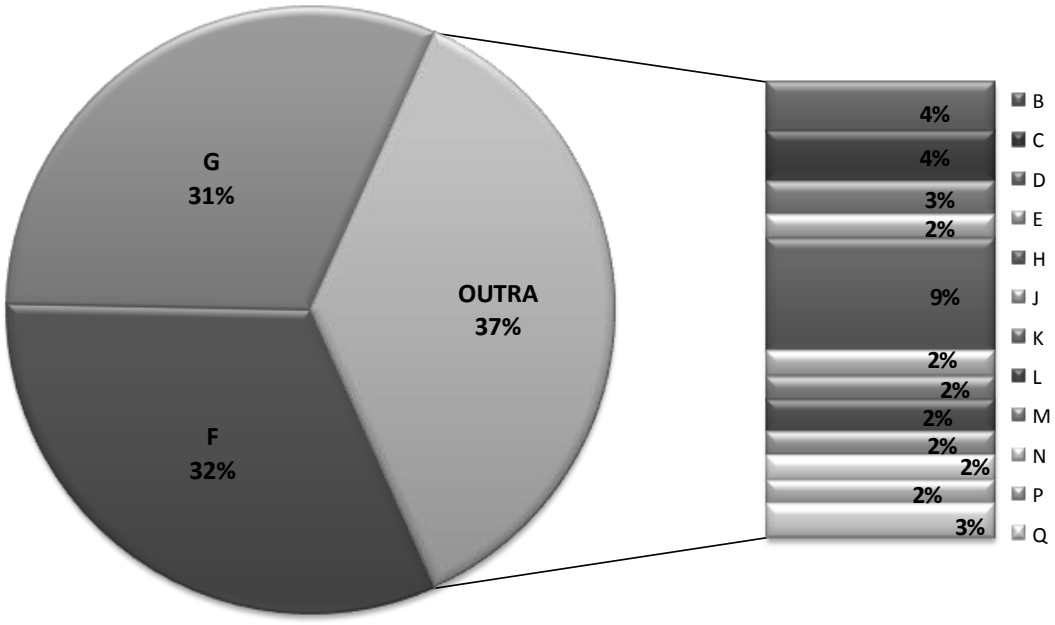


Figura 16: Distribuição dos documentos pelas Secções da CAE-Rev.3

Tabela 6: Distribuição dos documentos por categorias do 2º Nível

Categoria	Total de Documentos
41	86
42	83
43	87
45	85
46	85
47	81

No R, através do *package* 'tm', está disponível a função Corpus() para criar o Corpus da coleção de documentos (Feinerer *et al.*, 2008; Feinerer, 2011). Os comandos seguintes demonstram esse procedimento.

```
directorio <- DirSource(caminho)
documentos <- Corpus(directorio, readerControl = list(reader=readPlain, language='pt'))
documentos
A corpus with 800 text documents
```

```
Error: invalid input 'Em Março de 2004 abriu a loja de venda ao público e passou a disponibilizar todo o material informático, bem como todos os serviços técnicos associados.' in 'utf8towcs'
```

Figura 17: Exemplo de documento carregado no Corpus com erro

Como exemplo de *output* do Corpus tem-se a Figura 17, onde é possível observar que existe um problema de leitura devido à existência de caracteres inválidos. Para contornar este problema, todos os documentos são codificados em 'latin1', após serem lidos no R.

Latin1 trata-se de uma codificação de caracteres do alfabeto latino (romano) [4], sendo este o alfabeto utilizado para escrever a língua portuguesa e a maioria das línguas da Europa ocidental e central e das áreas colonizadas por europeus [3].

O código seguinte é referente a essa codificação:

```
for (i in seq_along(documentos))
        {
        Encoding(documentos[[i]]) <- "latin1"
        }
```

Após a codificação do exemplo apresentado na Figura 17 será obtido como *output* o documento apresentado como exemplo na Figura 2.

3.5. Preparação dos Dados

A etapa pré-processamento de texto desempenha um papel importante na classificação de documentos de texto. A ordem pela qual as tarefas de pré-processamento são executadas faz diferença na representação final dos documentos. Neste estudo consideram-se duas abordagens de preparação dos dados, como esquematizado na figura seguinte.

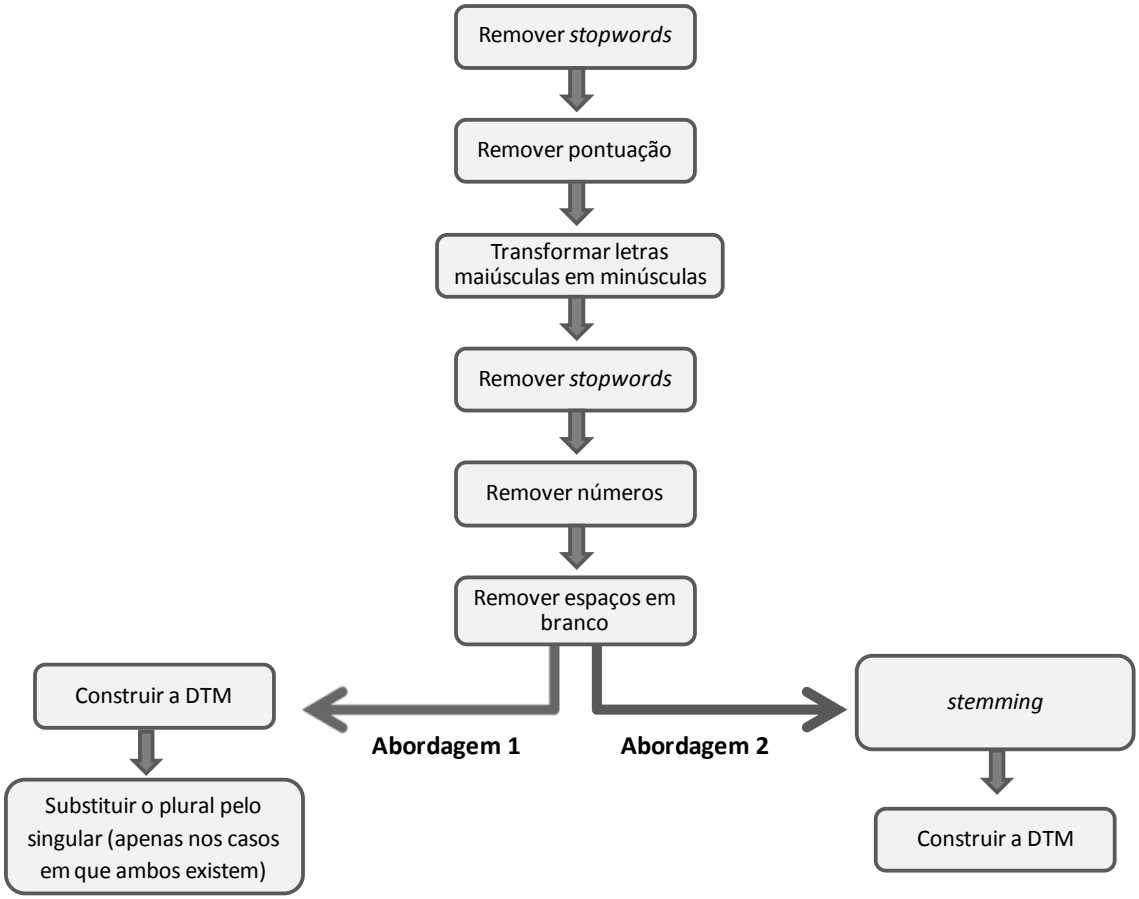


Figura 18: Tarefas de pré-processamento – duas abordagens

As duas abordagens de pré-processamento variam em duas tarefas. Na primeira abordagem é efetuada uma substituição dos plurais pelos singulares, enquanto na segunda abordagem é efetuado *stemming*.

O R tem funções, disponibilizadas através do *package* 'tm', que permitem efetuar as tarefas descritas anteriormente. Após criar o Corpus da coleção de documentos (documentos), são aplicados os comandos apresentados de seguida para as diversas tarefas de pré-processamento a todo o Corpus.

- Remover as *stopwords*:

```
docs_proc <- tm_map(documentos, removeWords, stopwords(language="pt"))
```

- Remover a pontuação:

```
docs_proc <- tm_map(docs_proc, removePunctuation)
```

- Transformar as letras maiúsculas em minúsculas:

```
docs_proc <- tm_map(docs_proc, tolower)
```

- Remover os números:

```
docs_proc <- tm_map(docs_proc, removeNumbers)
```

- Remover os espaços em branco:

```
docs_proc <- tm_map(docs_proc, stripWhitespace)
docs_proc <- tm_map(docs_proc, str_trim)
```

Nota: A função str_trim() depende da instalação do *package* 'stringr'.

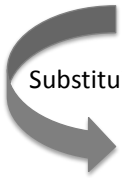
- Substituir o plural pelo singular:

O método usado para remover os plurais, consiste em selecionar a palavra candidata a plural e verificar se existe o singular dessa palavra. No caso de existir, agrupar a informação de ambas e considerar, como termo, a palavra no singular. Apenas foram programadas as regras mais frequentes, apresentadas de seguida:
Regra 1: Palavras que terminam em 's' e a penúltima letra é uma vogal;
Regra2: Palavras que terminam em 'es' e a antepenúltima letra pertence a ('r','z','n');
Regra 3: Palavras que terminam em 'is' e ao substituir 'is' por 'l' a palavra termina em ('al','el','ol','ul');
Regra 4: Palavras que terminam em 'is' e ao substituir 's' por 'l' a palavra termina em 'il';
Regra 5: Palavras que terminam em 'ão'.

A Tabela 7 apresenta um exemplo da aplicação deste método e no Anexo 2 é possível visualizar o código em R.

Tabela 7: Exemplo do método de substituição do plural pelo singular

	empresa	empresas	construção	construções	mensal	mensais	par	pares	funil	funis
d1	1	0	0	1	1	1	0	0	0	1
d2	1	0	1	1	1	1	1	0	0	0
d3	0	0	1	1	0	1	1	0	0	0
d4	1	0	1	0	0	1	1	1	1	1
d5	1	0	1	0	1	1	0	0	0	0



	empresa	construção	mensal	par	funil
d1	1	1	2	0	1
d2	1	2	2	1	0
d3	0	2	1	1	0
d4	1	1	1	2	2
d5	1	1	2	0	0

- *Stemming*:

O R tem disponível alguns *packages* com funções para executar a tarefa *stemming*. No entanto, como referido no capítulo 2, as funções não se adaptam bem à língua portuguesa, por não terem sido implementadas nesta língua.

Aplicadas algumas funções disponíveis no R, a uma amostra de palavras (ver Anexo 3), optou-se por usar a função stemDocument().

```
docs_stem<- tm_map(docs_proc, stemDocument)
```

- DTM:

```
DocumentTermMatrix(docs_proc)
```

As Figuras 19 e 20 permitem ter uma noção do *output* sem *stemming* (primeira abordagem) e com *stemming* (segunda abordagem), respetivamente.

	cliente	construção	mercado	produto	qualidade
001	0	0	0	2	0
002	2	0	0	1	0
003	0	0	1	1	0
004	5	0	4	3	2
005	5	0	0	1	2
006	0	0	0	0	1
007	3	0	1	3	4
008	4	0	2	1	0
009	2	0	0	1	2
010	9	0	1	0	0

Figura 19: Exemplo de representação dos dados após a primeira abordagem (sem *stemming*)

	client	construçã	merc	produt	qualidad
001	0	0	0	2	0
002	2	0	0	1	0
003	0	0	1	1	0
004	5	0	4	3	2
005	5	0	0	1	2
006	0	0	0	0	1
007	3	0	1	3	4
008	4	0	2	1	0
009	2	0	0	1	2
010	9	0	1	0	0

Figura 20: Exemplo de representação dos dados após a segunda abordagem (com *stemming*)

Inicialmente, o número total de termos é 14817. Após a aplicação das tarefas de pré-processamento mencionadas na abordagem 1, o número de termos fica reduzido a 8703, enquanto após a aplicação das tarefas de pré-processamento mencionadas na abordagem 2, permanecem apenas 6256 termos.

Neste estudo, sempre que se proceder à transformação da coleção de documentos numa DTM, consideram-se os seguintes filtros:

⇒ `weighting = weightTf`

⇒ `minWordLength = 2`

```
dtm_tot <- DocumentTermMatrix(tot_docs,control=list(
            minWordLength = 2, weighting = weightTf))
```

3.6. Conjunto de Treino e Conjunto de Teste

O conjunto de documentos é dividido em dois conjuntos, um conjunto de treino para induzir o classificador e um conjunto de teste para avaliar o desempenho do classificador. O conjunto de treino e o conjunto de teste foram selecionados através do método de amostragem *Holdout* estratificada (a distribuição de categoria é considerada durante a seleção). Este método garante uma seleção de documentos cuja dimensão por categoria é proporcional à dimensão das categorias na coleção de documentos em estudo.

Método de seleção:

⇒ Selecionar aleatoriamente 70% dos dados para treino de cada uma das categorias do nível mais desagregado. Na Figura 6 pode observar-se que ao nível mais desagregado estão representadas as categorias ‘41’, ‘42’, ‘43’, ‘45’, ‘46’, ‘47’ e ‘OUTRA’. No caso da categoria ‘OUTRA’, considera-se 70% de documentos de cada uma das Secções que a constituem (ver Figura 16).

Assim:

- 70% dos documentos são selecionados aleatoriamente de cada uma das categorias ao nível mais desagregado.
- Os restantes 30% dos documentos vão constituir o conjunto de teste.

No R, os conjuntos são obtidos, através da aplicação de alguns comandos explicados de seguida.

Começa-se por selecionar as posições dos documentos pertencentes às diferentes categorias. Exemplifica-se este procedimento com a categoria ‘41’.

```
pos_textos_F_41 <- which(empresas_CAE$DIV=="41")
```

O comando anterior devolve a posição em que se encontram os documentos da categoria ‘41’ no ficheiro ‘empresas_CAE’. A informação contida neste ficheiro pode

ser observada parcialmente no Anexo 4. Após ter as posições em que se encontram os documentos é feita um seleção aleatória de aproximadamente 70% dos mesmos através do comando seguinte:

```
s41 <- sample(1:length(pos_textos_F_41),round(length(pos_textos_F_41)*0.7)))
```

Após aplicar este procedimento para as categorias mais desagregadas da hierarquia considerada neste estudo, obtém-se o número de documentos para treino e para teste observados na Tabela 8.

De seguida são apresentados os procedimentos para obter os documentos do conjunto de treino e os documentos do conjunto de teste, de cada uma das categorias do primeiro nível de classificação ('F', 'G', e 'OUTRA').

Documentos de treino:

```
cl_F.train <- c(docs_proc[pos_textos_F_41[s41]],
                docs_proc[pos_textos_F_42[s42]],
                docs_proc[pos_textos_F_43[s43]])
cl_G.train <- c(docs_proc[pos_textos_G_45[s45]],
                docs_proc[pos_textos_G_46[s46]],
                docs_proc[pos_textos_G_47[s47]])
cl_OUTRA.train <- c(docs_proc[pos_textos_B[sB]],  docs_proc[pos_textos_C[sC]],
                docs_proc[pos_textos_D[sD]],  docs_proc[pos_textos_E[sE]],
                docs_proc[pos_textos_H[sH]],  docs_proc[pos_textos_J[sJ]],
                docs_proc[pos_textos_K[sK]], docs_proc[pos_textos_L[sL]],
                docs_proc[pos_textos_M[sM]], docs_proc[pos_textos_N[sN]],
               docs_proc[pos_textos_P[sP]], docs_proc[pos_textos_Q[sQ]])
```

Documentos de teste:

```
cl_F.test <- c(docs_proc[pos_textos_F_41[-s41]],
                docs_proc[pos_textos_F_42[-s42]],
                docs_proc[pos_textos_F_43[-s43]])
```

```
cl_G.test <- c(docs_proc[pos_textos_G_45[-s45]],
               docs_proc[pos_textos_G_46[-s46]],
               docs_proc[pos_textos_G_47[-s47]])
cl_OUTRA.test <- c(docs_proc[pos_textos_B[-sB]], docs_proc[pos_textos_C[-sC]],
               docs_proc[pos_textos_D[-sD]], docs_proc[pos_textos_E[-sE]],
               docs_proc[pos_textos_H[-sH]], docs_proc[pos_textos_J[-sJ]],
               docs_proc[pos_textos_K[-sK]], docs_proc[pos_textos_L[-sL]],
               docs_proc[pos_textos_M[-sM]], docs_proc[pos_textos_N[-sN]],
               docs_proc[pos_textos_P[-sP]], docs_proc[pos_textos_Q[-sQ]])
```

Número de documentos, por categoria, nos respetivos conjuntos.

Treino:

```
l1.train <- length(cl_F.train)
l2.train <- length(cl_G.train)
l3.train <- length(cl_OUTRA.train)
```

Teste:

```
l1.test <- length(cl_F.test)
l2.test <- length(cl_G.test)
l3.test <- length(cl_OUTRA.test)
```

- Anexar a informação da categoria ao *data.frame* - dtm

➢ Passo1: gerar um vetor com a informação da categoria

```
CLASS <- as.factor(ifelse(empresas_CAE$SEC=='F', 'F',
                          ifelse(empresas_CAE$SEC=='G', 'G', 'OUTRA')))
```

➢ Passo2: anexar este vetor à última coluna do *data.frame*

```
dtm$CLASS <-CLASS
last.col <- length(dtm)
```

Após estes procedimentos está reunida a informação necessária para a construção do *data.frame* para treino (dtm.train) e do *data.frame* para teste (dtm.test).

- Construir o conjunto de treino

```
dtm.train <- dtm[sort(substr(names(c(cl_F.train, cl_G.train, cl_OUTRA.train)),1,3)),]
```

- Construir o conjunto de teste

```
dtm.test <- dtm[ sort( substr( names(c(cl_F.test, cl_G.test, cl_OUTRA.test)), 1, 3)),
        1:(last.col-1)]
```

Estes procedimentos mostraram como são obtidos os conjuntos de treino e teste no primeiro nível da classificação, independentemente do tipo de problema de classificação ser classificação local por nó pai ou classificação local por nível. O número de documentos em ambos os conjuntos pode ser observado na Tabela 8.

Tabela 8: Número de documentos selecionados para o conjunto de treino e conjunto de teste no momento inicial

		Número documentos	Número documentos	Conjunto de Teste	Conjunto de Treino
Categoria 'F'	**41**	60	26	179	77
	42	58	25		
	43	61	26		
Categoria 'G'	**45**	59	26	175	76
	46	59	26		
	47	57	24		
Categoria 'OUTRA'	**B**	22	10	202	91
	C	22	10		
	D	15	7		
	E	10	5		
	H	50	22		
	J	11	5		
	K	10	5		
	L	15	6		
	M	10	5		
	N	11	5		
	P	11	5		
	Q	15	6		
	Total	**556**	**244**	**556**	**244**

O número de documentos no conjunto de teste considerando a classificação no segundo nível difere segundo o tipo de problema de classificação que se está a considerar, como se verificará no capítulo seguinte.

3.7. Seleção de Características

A seleção das características relevantes é um processo importante na redução da dimensão do conjunto de treino. Após a seleção de características relevantes, tem-se como resultado um conjunto de documentos de treino "limpo" que pode ser usado para uma classificação eficaz (Han e Kamber, 2006).

A seleção dos termos relevantes é importante para a criação do modelo. O ganho de Informação avalia o valor de um termo, medindo o ganho de informação com respeito à categoria.

Neste ponto será usada a função info() que calcula a informação do termo *i* e a função find.info.terms() que seleciona os termos com maior informação (Brazdil, 2009). As funções podem ser visualizadas no Anexo 5.

3.8. Classificadores

Os classificadores utilizadores neste estudo foram os descritos no capítulo 2. De seguida é exemplificado como são construídos no R.

Considere-se:

- → dtm.train representa o conjunto de treino;
- → dtm.test representa o conjunto de teste;
- → class.test representa os valores verdadeiros das categorias;
- → clas.formula representa os termos em função da categoria.

- Árvores de Decisão

Para obter uma árvore de decisão no R é necessário carregar o *package* 'rpart'.

```
library(rpart)
```

A função que produz a árvore de decisão é a função rpart()

```
dt <- rpart(clas.formula, dtm.train)
```

Os valores previstos são obtidos com o seguinte comando:

```
preds.dt <- predict(dt, dtm.test, type="class")
```

A matriz de confusão é obtida através da construção de uma tabela entre os valores reais e os valores preditos.

```
conf.mx.dt <- table(class.test, preds.dt)
```

- K-NN

Para o classificador *k*-NN é necessário carregar o *package* 'class'.

```
library(class)
```

Os valores previstos são obtidos com o seguinte comando:

```
preds.knn <- knn(dtm.train[,info.terms], dtm.test[,info.terms], class.train, k=13)
```

Matriz de confusão:

```
conf.mx.knn <- table(class.test, preds.knn)
```

- Redes Neuronais

Para obter uma rede neuronal é necessário carregar o *package* 'nnet'.

```
library(nnet)
```

A função que produz o classificador rede neuronal é a função nnet().

```
rn.classifier <- nnet(clas.formula, data=dtm.train, size=2, rang=0.1, decay=5e-4, maxit=200, MaxNWts = 1100)
```

Comandos para obter os valores previstos e matriz de confusão:

```
preds.rn <- predict(rn.classifier, dtm.test, type="class")
conf.mx.rn <- table(class.test, preds.rn)
```

- *Naive Bayes*

O classificador *Naive Bayes* é construído através da função make_Weka_classifier() disponível no *package* 'RWeka'.

```
library(RWeka)
nb.classifier <- NB(clas.formula, dtm.train)
```

Comandos para obter os valores previstos e matriz de confusão:

```
preds.nb <- predict(nb.classifier, dtm.test)
conf.mx.nb <- table(class.test, preds.nb)
```

- SVM

Para construir uma SVM no R é necessário carregar o package 'e1071'.

```
library(e1071)
```

A função que constrói a SVM é a função svm().

```
svm.classifier <- svm(clas.formula, dtm.train)
```

Comandos para obter os valores previstos e matriz de confusão:

```
preds.svm <- predict(svm.classifier, dtm.test)
conf.mx.svm <- table(class.test, preds.svm)
```

- SVM Linear

Procedimento idêntico à SVM, no entanto, é necessário mudar o tipo de *kernel* para linear.

```
svm.classifier2 <- svm(clas.formula, dtm.train, kernel = "linear", cost = 50)
```

Os resultados obtidos são apresentados no capítulo seguinte.

3.9. Similaridade

A análise de similaridade a ser efetuada é entre documentos com o descritivo das empresas e documentos com o descritivo das categorias. Pretende-se, para cada documento descritivo da empresa, procurar a descrição da categoria CAE-Rev.3 mais similar a esse documento. Este método representa um método alternativo de classificação e pretende-se verificar se os resultados são competitivos em comparação com o método de classificação.

Apenas serão considerados os documentos das empresas tidos como exemplo no conjunto de treino. Os documentos com o descritivo das categorias correspondem à descrição da mesma na CAE-Rev.3. Por exemplo, o documento com a descrição da categoria 'F' pode ser observado no Anexo 6.

Os documentos com a descrição das categorias serão preparados segundo as duas abordagens descritas no subcapítulo 3.5. A análise será efetuada considerando as duas abordagens, sem e com *stemming* e, também, os dois níveis de classificação. Assim, numa primeira situação será obtida a tabela de proximidade entre os documentos das empresas (no conjunto de treino) e os documentos com o descritivo das categorias 'F', 'G' e 'OUTRA'. Na segunda situação será obtida a tabela de proximidade entre os documentos das empresas (no conjunto de treino) e os documentos com o descritivo das categorias '41', '42', '43', '45', '46' e '47'.

Os seguintes comandos demostram como é obtida tabela de proximidade, após serem efetuadas as tarefas de pré-processamento. Começa-se por juntar todos os documentos (descritivo empresas e descritivo categorias).

```
tot_docs <- c(empresas,categorias1)
```

A matriz Dissimilaridade Cosseno é obtida através dos seguintes comandos:

```
dis <- dissimilarity(dtm_tot, method = "cosine")
matriz_dis <- as.matrix(dis)
```

Calcular a similaridade (similaridade = 1- dissimilaridade)

```
costable <- (1-matriz_dis)
```

O comando seguinte permite obter a Tabela de Proximidade com 4 casas decimais.

```
simil_cos <- round(costable[1:length(empresas),(length(empresas)+1):length(tot_docs)],4)
```

Os resultados são apresentados no próximo capítulo.

CAPÍTULO 4

Resultados

Neste capítulo é efetuada uma análise dos resultados obtidos a partir das técnicas de referidas anteriormente. Começa-se por mostrar os resultados obtidos para as medidas de avaliação da performance dos diferentes classificadores usados neste estudo. Os resultados são mostrados para os dois tipos de problemas de classificação: classificação local por nó pai e classificação local por nível consideradas neste estudo, em ambas as situações também serão consideradas as duas abordagens tidas em conta no pré-processamento dos dados (com e sem *stemming*). A análise de similaridade entre os documentos com o descritivo das empresas e os documentos com o descritivo das categorias, também considera os documentos com e sem *stemming* e será feita em dois níveis. Finalmente serão discutidos os resultados obtidos.

4.1. Performance dos Classificadores

A performance dos classificadores é avaliada através das medidas descritas no capítulo 2. Estas medidas são referidas, na literatura, como medidas de performance de classificadores para problemas de classificação plana. No entanto, como em cada nível de classificação, o problema de classificação pode ser visto como um problema de classificação plana, optou-se por se considerar estas medidas de avaliação na avaliação da performance dos classificadores. Dumais e Chen (2000) usam as medidas de avaliação de classificadores usadas na classificação plana, no estudo que apresentaram sobre classificação hierárquica de conteúdos da *web*.

A função Evaluating_Classifer(), foi construída no R para calcular as medidas de avaliação da performance do algoritmo consideradas no capítulo 2. Esta função tem como parâmetro de entrada a matriz de confusão e o *output* obtido é o valor de cada uma das medidas consideradas. O código da função pode ser visto no Anexo 7.

4.1.1. Classificação no 1º nível

No primeiro nível os documentos podem ser classificados nas categorias 'F', 'G' e 'OUTRA'. Os dois tipos de problemas de classificação considerados neste estudo, classificação local por nó pai e classificação local por nível, têm em comum a classificação no primeiro nível. A Figura 21 ilustra esse processo.

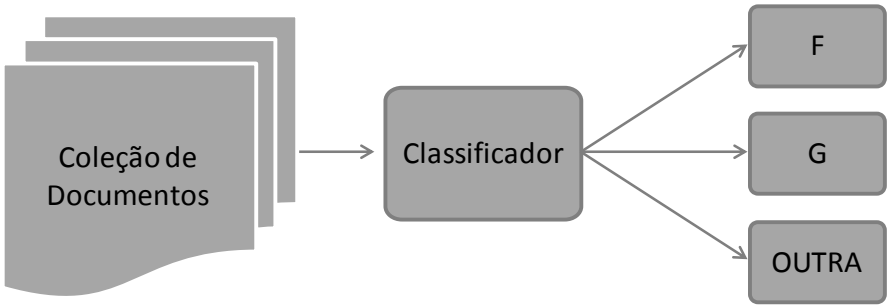


Figura 21: Esquema da classificação de documentos no 1º nível

O número de documentos de texto usados para treino e para teste é referido na tabela seguinte. O método de seleção foi o descrito no subcapítulo 3.6.

Tabela 9: Número de documentos no conjunto de treino e no conjunto de teste, nas diferentes abordagens (classificação local por nó pai e classificação local por nível), para o primeiro nível de classificação

		Conjunto de Treino	Conjunto de Teste
1º Nível	F	179	77
	G	175	76
	OUTRA	202	91
	TOTAL	**556**	**244**

Como pode ser observado na Tabela 9, em ambas as abordagens o classificador aprende com 556 exemplos de documentos de empresas e é avaliado através de 244 exemplos. Como já foi referido, os classificadores têm o mesmo procedimento no primeiro nível de classificação.

Após serem selecionados os conjuntos de treino e teste, é feita uma seleção dos termos relevantes. Essa seleção é feita através da função disponível no Anexo 5 e explicada no subcapítulo 3.7.

```
find.info.terms(dtm.train,0.005)
```

Com o valor mínimo de informação considerado (0.005) são mantidos 1629 termos no conjunto de treino composto por exemplos de documentos sem *stemming* e 1407 termos se os exemplos forem com *stemming*. Nas Figuras 22 e 23 podem ser observadas as matrizes de confusão geradas para cada um dos classificadores, considerando as duas abordagens referidas nas tarefas de pré-processamento, sem *stemming* e com *stemming*, respetivamente.

dt

Classes Previstas \ Classes Reais	F	G	OUTRA
F	53	1	23
G	5	49	22
OUTRA	7	19	65

KNN

Classes Previstas \ Classes Reais	F	G	OUTRA
F	71	0	6
G	18	43	15
OUTRA	12	17	62

SVM

Classes Previstas \ Classes Reais	F	G	OUTRA
F	56	8	13
G	10	54	12
OUTRA	5	10	76

SVM Linear

Classes Previstas \ Classes Reais	F	G	OUTRA
F	55	6	16
G	6	64	6
OUTRA	7	10	74

RN

Classes Previstas \ Classes Reais	F	G	OUTRA
F	68	3	6
G	2	67	7
OUTRA	12	39	40

NB

Classes Previstas \ Classes Reais	F	G	OUTRA
F	66	4	7
G	6	66	4
OUTRA	10	15	66

Figura 22: Matrizes de confusão: primeiro nível de classificação (documentos sem *stemming*)

dt

Classes Previstas \ Classes Reais	F	G	OUTRA
F	63	4	10
G	13	47	16
OUTRA	15	14	62

KNN

Classes Previstas \ Classes Reais	F	G	OUTRA
F	72	0	5
G	18	42	16
OUTRA	32	11	48

SVM

Classes Previstas \ Classes Reais	F	G	OUTRA
F	61	6	10
G	10	58	8
OUTRA	7	11	73

SVM Linear

Classes Previstas \ Classes Reais	F	G	OUTRA
F	61	6	10
G	6	61	9
OUTRA	7	14	70

RN

Classes Previstas \ Classes Reais	F	G	OUTRA
F	30	24	23
G	2	51	23
OUTRA	3	10	78

NB

Classes Previstas \ Classes Reais	F	G	OUTRA
F	67	4	6
G	7	64	5
OUTRA	12	13	66

Figura 23: Matrizes de confusão: primeiro nível de classificação (documentos com *stemming*)

A tabela seguinte apresenta os resultados das medidas de performance dos classificadores, na classificação no primeiro nível.

Tabela 10: Resultados das medidas de avaliação da performance dos classificadores, relativas às categorias 'F', 'G e 'OUTRA', considerando os documentos com *stemming* e sem *stemming*

	Sem *Stemming*						**Com** *Stemming*					
	Árvore Decisão	k-NN (k=11)	SVM	SVM Linear	Rede Neuronal	Naive Bayes	Árvore Decisão	k-NN (k=11)	SVM	SVM Linear	Rede Neuronal	Naive Bayes
Error_Rate	31,56	27,87	23,77	20,90	28,28	**18,85**	29,51	33,61	21,31	21,31	34,84	**19,26**
Precision_F	0,8154	0,7030	0,7887	0,8088	0,8293	0,8049	0,6923	0,5902	0,7821	0,8243	0,8571	0,7791
Precision_G	0,7101	0,7167	0,7500	0,8000	0,6147	0,7765	0,7231	0,7925	0,7733	0,7531	0,6000	0,7901
Precision_OUTRA	0,5909	0,7470	0,7525	0,7708	0,7547	0,8571	0,7045	0,6957	0,8022	0,7865	0,6290	0,8571
Recall_F	0,6883	0,9221	0,7273	0,7143	0,8831	0,8571	0,8182	0,9351	0,7922	0,7922	0,3896	0,8701
Recall_G	0,6447	0,5658	0,7105	0,8421	0,8816	0,8684	0,6184	0,5526	0,7632	0,8026	0,6711	0,8421
Recall_OUTRA	0,7143	0,6813	0,8352	0,8132	0,4396	0,7253	0,6813	0,5275	0,8022	0,7692	0,8571	0,7253
F1_F	0,7465	0,7978	0,7568	0,7586	0,8553	0,8302	0,7500	0,7236	0,7871	0,8079	0,5357	0,8221
F1_G	0,6759	0,6324	0,7297	0,8205	0,7243	0,8199	0,6667	0,6512	0,7682	0,7771	0,6335	0,8153
F1_OUTRA	0,6468	0,7126	0,7917	0,7914	0,5556	0,7857	0,6927	0,6000	0,8022	0,7778	0,7256	0,7857
Macro_F1	0,6938	0,7226	0,7607	0,7915	0,7338	0,8149	0,7063	0,6821	0,7859	0,7880	0,6662	0,8106
Micro_F1	0,6844	0,7213	0,7623	0,7910	0,7172	0,8115	0,7049	0,6639	0,7869	0,7869	0,6516	0,8074

O algoritmo que obteve o melhor desempenho foi o *Naïve Bayes*, tanto no caso em que os documentos passaram pela tarefa de *stemming* como no caso em que esta tarefa não é aplicada. O caso em que o algoritmo obteve a menor taxa de erro foi na classificação de documentos sem *stemming*, obteve uma taxa de erro de 18.85 que pode ser considerada bastante boa.

Assim, os exemplos considerados para o conjunto de teste do nível seguinte são os classificados corretamente neste nível pelo classificador *Naïve Bayes*.

4.1.2. Classificador no 2º Nível

A classificação no segundo nível vai ser efetuada considerando dois métodos diferentes. Classificação local por nó pai e classificação local por nível.

- **Classificação local por nó pai**

Na classificação local por nó pai, se o classificador do primeiro nível atribuiu o exemplo à categoria 'G', o classificador do segundo nível só treina com as categorias

‘45’, ‘46’ e ‘47’. Se o classificador do primeiro nível atribuiu o exemplo à categoria ‘F’, o classificador do segundo nível só treina com as categorias ‘41’, ‘42’, e ‘43’. Assim, no caso em estudo, a classificação no segundo nível, considerando a abordagem classificação local por nó pai, vai ter dois classificadores, um para classificar os filhos de ‘F’ e outro para classificar os filhos de ‘G’.

A Tabela 11 apresenta o número de documentos no conjunto de treino e no conjunto de teste, considerando documentos com e sem *stemming*. Como referido no tópico anterior, os exemplos considerados para o conjunto de teste são os classificados corretamente no primeiro nível de classificação. Para melhor exemplificar, considere-se, a título de exemplo, o caso sem *stemming*. O classificador *Naive Bayes* foi o que obteve a melhor performance no primeiro nível. Pela sua matriz de confusão apresentada na Figura 23, pode-se observar que 66 documentos foram classificados corretamente na categoria F, assim, apenas esses documentos serão considerados no conjunto de teste, na classificação do segundo nível, para o classificador que irá classificar os ‘filhos’ de F.

Tabela 11: Número de documentos no conjunto de treino e no conjunto de teste, para o segundo nível de classificação, considerando a classificação local por nó pai

		Classificação local por nó pai			
		SEM Stemming		**COM Stemming**	
		Conjunto de Treino	**Conjunto de Teste**	**Conjunto de Treino**	**Conjunto de Teste**
2º Nível	**41**	60	23	60	21
	42	58	21	58	23
	43	61	22	61	23
	TOTAL F	**179**	**66**	**179**	**67**
	45	59	26	59	24
	46	59	19	59	18
	47	57	21	57	22
	TOTAL G	**175**	**66**	**175**	**64**

➢ <u>Primeiro Classificador</u>: classificar os ‘filhos’ de F

A Figura 24 esquematiza o processo de classificação deste classificador.

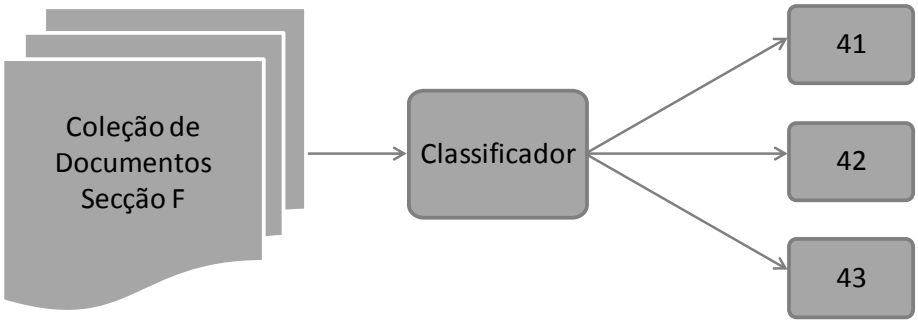


Figura 24: Esquema da classificação de texto no 2º nível - categorias descendentes de 'F'

Considerações:

⇒ O conjunto de teste contém apenas os documentos da categoria 'F' classificados corretamente no nível anterior pelo algoritmo *Naive Bayes*

⇒ find.info.terms(dtm.train,0.01)

Com o valor mínimo de informação considerado (0.01) são mantidos 853 termos no conjunto de treino composto por exemplos de documentos sem *stemming* e 773 termos se os exemplos forem com *stemming*. Nas Figuras 25 e 26 podem ser observadas as matrizes de confusão geradas para cada um dos classificadores, considerando as duas abordagens referidas nas tarefas de pré-processamento, sem *stemming* e com *stemming*, respetivamente.

dt

		Classes Reais		
		41	**42**	**43**
Classes Previstas	**41**	14	2	7
	42	5	9	7
	43	3	4	15

KNN

		Classes Reais		
		41	**42**	**43**
Classes Previstas	**41**	3	17	3
	42	1	17	3
	43	0	8	14

SVM

		Classes Reais		
		41	**42**	**43**
Classes Previstas	**41**	6	12	5
	42	10	9	2
	43	2	0	20

SVM Linear

		Classes Reais		
		41	**42**	**43**
Classes Previstas	**41**	11	11	1
	42	4	14	3
	43	1	4	17

RN

		Classes Reais		
		41	**42**	**43**
Classes Previstas	**41**	10	11	2
	42	5	14	2
	43	2	5	15

NB

		Classes Reais		
		41	**42**	**43**
Classes Previstas	**41**	16	7	0
	42	4	12	5
	43	6	4	12

Figura 25: Matrizes de confusão: segundo nível de classificação – classificação por nó F (documentos sem *stemming*)

dt

Classes Previstas	Classes Reais 41	42	43
41	12	3	6
42	6	11	6
43	2	5	16

KNN

Classes Previstas	Classes Reais 41	42	43
41	1	17	3
42	2	16	5
43	0	4	19

SVM

Classes Previstas	Classes Reais 41	42	43
41	8	7	6
42	11	11	1
43	4	1	18

SVM Linear

Classes Previstas	Classes Reais 41	42	43
41	7	12	2
42	7	14	2
43	3	5	15

RN

Classes Previstas	Classes Reais 41	42	43
41	17	1	3
42	13	8	2
43	5	3	15

NB

Classes Previstas	Classes Reais 41	42	43
41	16	4	1
42	6	14	3
43	5	7	11

Figura 26: Matrizes de confusão: segundo nível de classificação – classificação por nó F (documentos com *stemming*)

A tabela seguinte apresenta os resultados das medidas de performance dos classificadores, na classificação no segundo nível, local por nó F.

Tabela 12: Resultados das medidas de avaliação da performance dos classificadores, na classificação local por nó F no segundo nível, considerando os documentos com *stemming* e sem *stemming*

	Sem *Stemming*						**Com** *Stemming*					
	Árvore Decisão	k-NN (k=11)	SVM	SVM Linear	Rede Neuronal	Naive Bayes	Árvore Decisão	k-NN (k=11)	SVM	SVM Linear	Rede Neuronal	Naive Bayes
Error_Rate	42,42	48,48	46,97	**36,36**	40,91	39,39	41,79	46,27	44,78	46,27	40,30	**38,81**
Precision_41	0,6364	0,7500	0,3333	0,6875	0,5882	0,6154	0,6000	0,3333	0,3478	0,4118	0,4857	0,5926
Precision_42	0,6000	0,4048	0,4286	0,4828	0,4667	0,5217	0,5789	0,4324	0,5789	0,4516	0,6667	0,5600
Precision_43	0,5172	0,7000	0,7407	0,8095	0,7895	0,7059	0,5714	0,7037	0,7200	0,7895	0,7500	0,7333
Recall_41	0,6087	0,1304	0,2609	0,4783	0,4348	0,6957	0,5714	0,0476	0,3810	0,3333	0,8095	0,7619
Recall_42	0,4286	0,8095	0,4286	0,6667	0,6667	0,5714	0,4783	0,6957	0,4783	0,6087	0,3478	0,6087
Recall_43	0,6818	0,6364	0,9091	0,7727	0,6818	0,5455	0,6957	0,8261	0,7826	0,6522	0,6522	0,4783
F1_41	0,6222	0,2222	0,2927	0,5641	0,5000	0,6531	0,5854	0,0833	0,3636	0,3684	0,6071	0,6667
F1_42	0,5000	0,5397	0,4286	0,5600	0,5490	0,5455	0,5238	0,5333	0,5238	0,5185	0,4571	0,5833
F1_43	0,5882	0,6667	0,8163	0,7907	0,7317	0,6154	0,6275	0,7600	0,7500	0,7143	0,6977	0,5789
Macro_F1	0,5787	0,5681	0,5164	0,6494	0,6044	0,6092	0,5826	0,5059	0,5481	0,5410	0,6183	0,6224
Micro_F1	0,5758	0,5152	0,5303	0,6364	0,5909	0,6061	0,5821	0,5373	0,5522	0,5373	0,5970	0,6119

No caso dos da abordagem sem *stemming*, o classificador que produziu os melhores resultados foi o classificador SVM Linear. O algoritmo que obteve o melhor desempenho na abordagem com *stemming* foi o *Naïve Bayes*.

- **<u>Segundo Classificador: classificar os 'filhos' de G</u>**

A Figura 27 esquematiza o processo de classificação deste classificador.

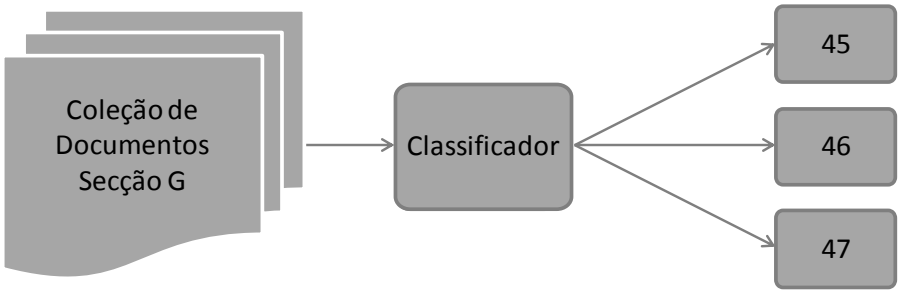


Figura 27: Esquema da classificação de texto no 2º nível - categorias descendentes de 'G'

Considerações:

⇒ O conjunto de teste contém apenas os documentos da categoria 'G' classificados corretamente no nível anterior pelo algoritmo *Naive Bayes*

⇒ find.info.terms(dtm.train,0.005)

Com o valor mínimo de informação considerado (0.01) são mantidos 919 termos no conjunto de treino composto por exemplos de documentos sem *stemming* e 862 termos se os exemplos forem com *stemming*. Nas Figuras 28 e 29 podem ser observadas as matrizes de confusão geradas para cada um dos classificadores, considerando as duas abordagens referidas nas tarefas de pré-processamento, sem *stemming* e com *stemming*, respetivamente.

dt

		Classes Reais		
		45	**46**	**47**
Classes Previstas	**45**	19	2	5
	46	0	10	9
	47	1	15	5

KNN

		Classes Reais		
		45	**46**	**47**
Classes Previstas	**45**	23	2	1
	46	11	6	2
	47	12	3	6

SVM

		Classes Reais		
		45	**46**	**47**
Classes Previstas	**45**	21	5	0
	46	7	10	2
	47	4	7	10

SVM Linear

		Classes Reais		
		45	**46**	**47**
Classes Previstas	**45**	24	2	0
	46	4	12	3
	47	2	6	13

RN

		Classes Reais		
		45	**46**	**47**
Classes Previstas	**45**	24	1	1
	46	2	10	7
	47	0	5	16

NB

		Classes Reais		
		45	**46**	**47**
Classes Previstas	**45**	25	0	1
	46	2	12	5
	47	1	4	16

Figura 28: Matrizes de confusão: segundo nível de classificação – classificação por nó G (documentos sem *stemming*)

dt

Classes Previstas	Classes Reais 45	46	47
45	15	3	6
46	1	9	8
47	1	9	12

KNN

Classes Previstas	Classes Reais 45	46	47
45	23	1	0
46	10	5	3
47	14	0	8

SVM

Classes Previstas	Classes Reais 45	46	47
45	17	7	0
46	5	10	3
47	5	6	11

SVM Linear

Classes Previstas	Classes Reais 45	46	47
45	20	4	0
46	1	12	5
47	2	6	14

RN

Classes Previstas	Classes Reais 45	46	47
45	11	12	1
46	2	12	4
47	2	9	11

NB

Classes Previstas	Classes Reais 45	46	47
45	22	2	0
46	2	11	5
47	2	2	18

Figura 29: Matrizes de confusão: segundo nível de classificação – classificação por nó G (documentos com *stemming*)

A tabela seguinte apresenta os resultados das medidas de performance dos classificadores, na classificação no segundo nível, local por nó G.

Tabela 13: Resultados das medidas de avaliação da performance dos classificadores, na classificação local por nó G no segundo nível, considerando os documentos com *stemming* e sem *stemming*

	Sem *Stemming*						**Com** *Stemming*					
	Árvore Decisão	k-NN (k=11)	SVM	SVM Linear	Rede Neuronal	Naive Bayes	Árvore Decisão	k-NN (k=11)	SVM	SVM Linear	Rede Neuronal	Naive Bayes
Error_Rate	48,48	46,97	37,88	25,76	24,24	**19,70**	43,75	43,75	40,63	28,13	46,88	**20,31**
Precision_45	0,9500	0,5000	0,6563	0,8000	0,9231	0,8929	0,8824	0,4894	0,6296	0,8696	0,7333	0,8462
Precision_46	0,3704	0,5455	0,4545	0,6000	0,6250	0,7500	0,4286	0,8333	0,4348	0,5455	0,3636	0,7333
Precision_47	0,2632	0,6667	0,8333	0,8125	0,6667	0,7273	0,4615	0,7273	0,7857	0,7368	0,6875	0,7826
Recall_45	0,7308	0,8846	0,8077	0,9231	0,9231	0,9615	0,6250	0,9583	0,7083	0,8333	0,4583	0,9167
Recall_46	0,5263	0,3158	0,5263	0,6316	0,5263	0,6316	0,5000	0,2778	0,5556	0,6667	0,6667	0,6111
Recall_47	0,2381	0,2857	0,4762	0,6190	0,7619	0,7619	0,5455	0,3636	0,5000	0,6364	0,5000	0,8182
F1_45	0,8261	0,6389	0,7241	0,8571	0,9231	0,9259	0,7317	0,6479	0,6667	0,8511	0,5641	0,8800
F1_46	0,4348	0,4000	0,4878	0,6154	0,5714	0,6857	0,4615	0,4167	0,4878	0,6000	0,4706	0,6667
F1_47	0,2500	0,4000	0,6061	0,7027	0,7111	0,7442	0,5000	0,4848	0,6111	0,6829	0,5789	0,8000
Macro_F1	0,5127	0,5304	0,6249	0,7310	0,7377	0,7875	0,5733	0,5990	0,6020	0,7147	0,5670	0,7847
Micro_F1	0,5152	0,5303	0,6212	0,7424	0,7576	0,8030	0,5625	0,5625	0,5938	0,7188	0,5313	0,7969

Neste caso, o classificador *Naïve Bayes* foi o que obteve a melhor performance nas duas abordagens.

- **Classificação local por nível**

O classificador local por nível consiste no treino de um classificador multi-classe para cada nível da hierarquia de classes. Neste tipo de classificação, são treinados dois classificadores, um classificador para cada nível. No primeiro nível o classificador será treinado para prever 3 categorias, enquanto no segundo nível será treinado para prever 6 categorias.

O esquema representativo do processo de classificação deste classificador pode ser visualizado na Figura 30.

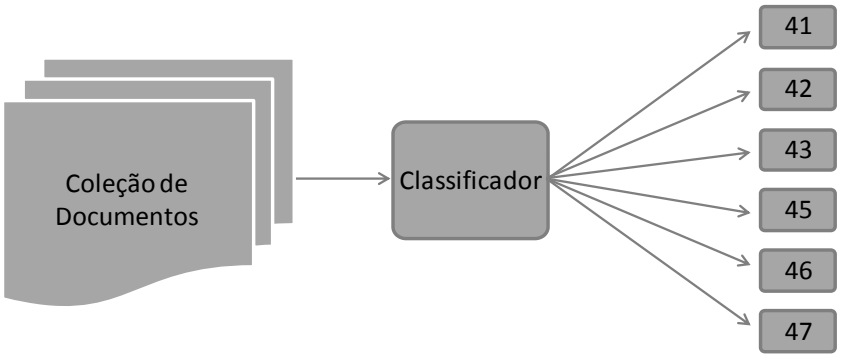


Figura 30: Esquema da classificação de texto ao nível da Divisão, considerando classificação local por nível

A Tabela 14 apresenta o número de documentos no conjunto de treino e no conjunto de teste considerando documentos com e sem *stemming*.

Tabela 14: Número de documentos no conjunto de treino e no conjunto de teste, para o segundo nível de classificação, considerando a classificação local por nível

		Classificação local por nível			
		SEM Stemming		COM Stemming	
		Conjunto de Treino	Conjunto de Teste	Conjunto de Treino	Conjunto de Teste
2º Nível	41	60	23	60	21
	42	58	21	58	23
	43	61	22	61	23
	45	59	26	59	24
	46	59	19	59	18
	47	57	21	57	22
	TOTAL	354	132	354	131

Considerações:

⇒ O conjunto de teste contém os documentos das classes 'F' e 'G' classificados corretamente no primeiro nível pelo algoritmo *Naive Bayes*.

⇒ find.info.terms(dtm.train,0.01)

Com o valor mínimo de informação considerado (0.01) são mantidos 1463 termos no conjunto de treino composto por exemplos de documentos sem *stemming* e 1302 termos se os exemplos forem com *stemming*. Nas Figuras 31 e 32 podem ser observadas as matrizes de confusão, considerando as abordagens, sem *stemming* e com *stemming*, respetivamente.

dt

		Classes Reais					
		41	**42**	**43**	**45**	**46**	**47**
Classes Previstas	**41**	17	2	1	0	2	1
	42	9	9	2	0	0	1
	43	4	4	10	0	2	2
	45	1	0	1	18	1	5
	46	4	0	0	0	2	13
	47	0	1	3	1	7	9

KNN

		Classes Reais					
		41	**42**	**43**	**45**	**46**	**47**
Classes Previstas	**41**	2	19	0	1	0	1
	42	1	19	1	0	0	0
	43	0	7	15	0	0	0
	45	0	2	1	22	0	1
	46	0	3	3	5	5	3
	47	0	4	4	3	5	5

SVM

		Classes Reais					
		41	**42**	**43**	**45**	**46**	**47**
Classes Previstas	**41**	7	9	9	0	1	0
	42	4	6	1	0	7	0
	43	1	1	20	0	0	0
	45	0	0	2	18	6	0
	46	0	0	6	1	10	2
	47	0	0	6	1	9	5

SVM Linear

		Classes Reais					
		41	**42**	**43**	**45**	**46**	**47**
Classes Previstas	**41**	7	11	4	0	1	0
	42	3	15	2	0	0	1
	43	1	4	17	0	0	0
	45	0	1	0	21	4	0
	46	2	1	1	2	9	4
	47	0	1	2	1	5	12

RN

		Classes Reais					
		41	**42**	**43**	**45**	**46**	**47**
Classes Previstas	**41**	11	10	0	2	0	0
	42	3	17	1	0	0	0
	43	5	3	1	9	3	1
	45	0	0	0	22	4	0
	46	1	0	1	9	3	5
	47	0	0	0	5	6	10

NB

		Classes Reais					
		41	**42**	**43**	**45**	**46**	**47**
Classes Previstas	**41**	17	5	0	0	1	0
	42	3	13	4	0	1	0
	43	5	6	11	0	0	0
	45	1	0	0	23	1	1
	46	0	0	1	2	11	5
	47	1	0	2	0	4	14

Figura 31: Matrizes de confusão: segundo nível de classificação – classificação por nível (documentos sem *stemming*)

dt

Classes Previstas \ Classes Reais	41	42	43	45	46	47
41	16	3	0	0	1	1
42	11	12	0	0	0	0
43	3	6	8	0	1	5
45	1	0	1	16	2	4
46	3	0	1	0	3	11
47	0	1	2	1	7	11

KNN

Classes Previstas \ Classes Reais	41	42	43	45	46	47
41	1	17	1	1	0	1
42	1	16	5	0	1	0
43	0	8	13	1	0	1
45	0	3	3	15	1	2
46	0	1	3	4	7	3
47	0	3	3	6	3	7

SVM

Classes Previstas \ Classes Reais	41	42	43	45	46	47
41	7	5	8	0	1	0
42	5	6	7	0	5	0
43	2	0	20	0	1	0
45	0	0	1	16	7	0
46	0	0	5	1	8	4
47	0	0	5	0	7	10

SVM Linear

Classes Previstas \ Classes Reais	41	42	43	45	46	47
41	8	9	4	0	0	0
42	4	16	3	0	0	0
43	3	4	14	0	2	0
45	1	1	1	17	3	1
46	1	0	3	2	8	4
47	0	1	2	0	6	13

NB

Classes Previstas \ Classes Reais	41	42	43	45	46	47
41	17	2	2	0	0	0
42	5	14	3	0	1	0
43	5	7	10	0	0	1
45	1	0	1	21	1	0
46	0	0	1	2	10	5
47	0	1	1	1	3	16

RN

Classes Previstas \ Classes Reais	41	42	43	45	46	47
41	2	5	8	1	1	4
42	4	5	11	0	0	3
43	1	3	17	0	0	2
45	0	0	5	6	7	6
46	1	1	0	2	12	2
47	1	2	1	2	10	6

Figura 32: Matrizes de confusão: segundo nível de classificação – classificação por nível (documentos com *stemming*)

A tabela seguinte apresenta os resultados das medidas de performance dos classificadores, na classificação no segundo nível, local por nível.

Tabela 15: Resultados das medidas de avaliação da performance dos classificadores, na classificação local por nível, no segundo nível da hierarquia, considerando os documentos com e sem *stemming*

	Sem *Stemming*						**Com** *Stemming*					
	Árvore Decisão	k-NN (k=5)	SVM	SVM Linear	Rede Neuronal	Naive Bayes	Árvore Decisão	k-NN (k=5)	SVM	SVM Linear	Rede Neuronal	Naive Bayes
Error_Rate	50,76	48,48	47,73	38,64	51,52	**32,58**	49,62	54,96	48,85	41,98	63,36	**32,82**
Precision_41	0,4857	0,6667	0,5833	0,5385	0,5500	0,6296	0,4706	0,5000	0,5000	0,4706	0,2222	0,6071
Precision_42	0,5625	0,3519	0,5625	0,4545	0,5667	0,5417	0,5455	0,3333	0,5455	0,5161	0,3125	0,5833
Precision_43	0,5882	0,6250	0,4545	0,6538	0,3333	0,6111	0,6667	0,4643	0,4348	0,5185	0,4048	0,5556
Precision_45	0,9474	0,7097	0,9000	0,8750	0,4681	0,9200	0,9412	0,5556	0,9412	0,8947	0,5455	0,8750
Precision_46	0,1429	0,5000	0,3030	0,4737	0,1875	0,6111	0,2143	0,5833	0,2759	0,4211	0,4000	0,6667
Precision_47	0,2903	0,5000	0,7143	0,7059	0,6250	0,7000	0,3438	0,5000	0,7143	0,7222	0,2609	0,7273
Recall_41	0,7391	0,0870	0,3043	0,3043	0,4783	0,7391	0,7619	0,0476	0,3333	0,3810	0,0952	0,8095
Recall_42	0,4286	0,9048	0,4286	0,7143	0,8095	0,6190	0,5217	0,6957	0,2609	0,6957	0,2174	0,6087
Recall_43	0,4545	0,6818	0,9091	0,7727	0,0455	0,5000	0,3478	0,5652	0,8696	0,6087	0,7391	0,4348
Recall_45	0,6923	0,8462	0,6923	0,8077	0,8462	0,8846	0,6667	0,6250	0,6667	0,7083	0,2500	0,8750
Recall_46	0,1053	0,2632	0,5263	0,4737	0,1579	0,5789	0,1667	0,3889	0,4444	0,4444	0,6667	0,5556
Recall_47	0,4286	0,2381	0,2381	0,5714	0,4762	0,6667	0,5000	0,3182	0,4545	0,5909	0,2727	0,7273
F1_41	0,5862	0,1538	0,4000	0,3889	0,5116	0,6800	0,5818	0,0870	0,4000	0,4211	0,1333	0,6939
F1_42	0,4865	0,5067	0,4865	0,5556	0,6667	0,5778	0,5333	0,4507	0,3529	0,5926	0,2564	0,5957
F1_43	0,5128	0,6522	0,6061	0,7083	0,0800	0,5500	0,4571	0,5098	0,5797	0,5600	0,5231	0,4878
F1_45	0,8000	0,7719	0,7826	0,8400	0,6027	0,9020	0,7805	0,5882	0,7805	0,7907	0,3429	0,8750
F1_46	0,1212	0,3448	0,3846	0,4737	0,1714	0,5946	0,1875	0,4667	0,3404	0,4324	0,5000	0,6061
F1_47	0,3462	0,3226	0,3571	0,6316	0,5405	0,6829	0,4074	0,3889	0,5556	0,6500	0,2667	0,7273
Macro_F1	0,4884	0,5297	0,5492	0,6121	0,4619	0,6668	0,5116	0,4634	0,5349	0,5809	0,3654	0,6688
Micro_F1	0,4924	0,5152	0,5227	0,6136	0,4848	0,6742	0,5038	0,4504	0,5115	0,5802	0,3664	0,6718

Mais uma vez, o algoritmo que obteve o melhor desempenho foi o *Naive Bayes*.

No subcapítulo 4.3 será apresentada uma discussão dos resultados obtidos.

4.2. Análise de Similaridade: Descritivo empresa *vs*. Descritivo categoria

4.2.1. Categorias do 1º nível

Juntam-se os documentos pretendidos:

→ Descritivo da empresa retirado da *web*

→ Descritivo das categorias consideradas neste estudo: ‘F’, ‘G’ e ‘OUTRA’

(OUTRA: 'B', 'C', 'D', 'E', 'H', 'J', 'K', 'L', 'M', 'N', 'P' e 'Q')

- **Abordagem 1** (documentos SEM *stemming*)

Começa-se por se considerar os documentos das empresas e os documentos das categorias, sem a tarefa *stemmimg*.

```
empresas <- emp_preproc
categorias <- cat_preproc_N1
```

Consideram-se apenas os documentos das empresas usados no conjunto de teste, no primeiro nível, para construir a tabela de proximidade.

```
docs_teste <- rownames(dtm.test_preproc)
docs_teste <- as.numeric(as.character(docs_teste))
empresas <- empresas[docs_teste]
```

A Tabela 16 apresenta o output parcial da tabela de proximidade entre os documentos com o descritivo das empresas e o descritivo das categorias retirado da CAE-Rev.3.

Tabela 16: Output parcial da tabela de proximidade, dos documentos das empresas com os documentos das categorias do primeiro nível, com as classificações previstas e reais.

	F	G	OUTRA	PREVISTO	REAL
1	0,058	0,011	0,050	F	OUTRA
2	0,010	0,064	0,071	OUTRA	G
3	0,023	0,035	0,047	OUTRA	OUTRA
4	0,008	0,001	0,021	OUTRA	OUTRA
5	0,040	0,108	0,046	G	G
6	0,069	0,002	0,018	F	OUTRA
7	0,011	0,037	0,055	OUTRA	G
8	0,000	0,028	0,031	OUTRA	G
9	0,000	0,024	0,030	OUTRA	OUTRA
10	0,013	0,176	0,026	G	G

A tabela com o cruzamento das categorias previstas e reais pode ser visualizada na Tabela 17.

Tabela 17: Resultados da classificação (primeiro nível) pela análise de proximidade – documentos sem *stemming*

		Categorias Reais		
		F	G	OUTRA
Categorias Previstas	F	63	13	28
	G	1	25	5
	OUTRA	13	38	58

Dos 244 documentos considerados, apenas **146** são classificados corretamente. A taxa de erro ronda os 40%.

- **Abordagem 2** (documentos COM *stemming*)

Começa-se por considerar os documentos das empresas e os documentos das categorias, **com** a tarefa *stemmimg*.

```
empresas <- emp_stem
categorias <- cat_stem_N1
```

Considerar apenas os documentos das empresas do conjunto de teste, usados no primeiro nível da classificação, para construir a tabela de proximidade.

```
docs_teste <- rownames(dtm.test_stem)
docs_teste <- as.numeric(as.character(docs_teste))
empresas <- empresas[docs_teste]
```

A Tabela 18 apresenta o *output* parcial da tabela de proximidade entre os documentos das empresas e o descritivo das categorias, documentos aos quais foi aplicada a tarefa *stemming*.

Tabela 18: Output parcial da tabela de proximidade, dos documentos das empresas com os documentos das categorias do primeiro nível, ambos pós *stemming*, com as classificações previstas e reais.

	F	G	OUTRA	PREVISTO	REAL
1	0,176	0,014	0,328	OUTRA	OUTRA
2	0,057	0,077	0,112	OUTRA	G
3	0,052	0,034	0,123	OUTRA	OUTRA
4	0,041	0,001	0,140	OUTRA	OUTRA
5	0,082	0,180	0,042	G	G
6	0,072	0,002	0,044	F	OUTRA
7	0,077	0,046	0,242	OUTRA	G
8	0,000	0,027	0,024	G	G
9	0,000	0,028	0,007	G	OUTRA
10	0,018	0,202	0,008	G	G

A Tabela 19 apresenta o resultado do cruzamento das categorias previstas e com as categorias reais.

Tabela 19: Resultados da classificação (primeiro nível) pela análise de proximidade – documentos com *stemming*

		Categorias Reais		
		F	G	OUTRA
Categorias Previstas	F	53	12	25
	G	1	24	11
	OUTRA	23	40	55

Dos 244 documentos apenas **132** são classificados corretamente. A taxa de erro ronda os 46%. Esta abordagem, considerando os documentos pós-*stemming*, revela resultados piores do que a abordagem anterior.

4.2.2. Categorias do 2º nível

Juntam-se os documentos pretendidos:

→ Descritivo da empresa retirado da *web*

→ Descritivo das categorias consideradas neste estudo: '41', '42', '43', '45', '46' e '47'.

- **Abordagem 1** (documentos SEM *stemming*)

Começa-se por se considerar os documentos das empresas e os documentos das categorias, sem a tarefa *stemmimg*.

```
empresas <- emp_preproc
categorias <- cat_preproc_N2
```

Consideram-se apenas os documentos das empresas usados no conjunto de teste para construir a tabela de proximidade.

```
docs_teste <- rownames(dtm.test2_preproc)
docs_teste <- as.numeric(as.character(docs_teste))
empresas <- empresas[docs_teste]
```

A Tabela 20 apresenta o output parcial da tabela de proximidade entre os documentos com o descritivo das empresas e o descritivo das categorias retirado da CAE-Rev.3.

Tabela 20: Output parcial da tabela de proximidade, dos documentos das empresas com os documentos das categorias do segundo nível, com as classificações previstas e reais.

	41	42	43	45	46	47	PREV	REAL
1	0,000	0,018	0,000	0,005	0,056	0,055	46	47
2	0,000	0,026	0,062	0,027	0,033	0,149	47	47
3	0,000	0,021	0,000	0,025	0,030	0,032	47	46
4	0,000	0,000	0,000	0,000	0,027	0,021	46	47
5	0,000	0,000	0,038	0,094	0,192	0,103	46	47
6	0,037	0,114	0,065	0,000	0,013	0,005	42	46
7	0,000	0,000	0,009	0,033	0,009	0,017	45	46
8	0,000	0,044	0,000	0,046	0,042	0,033	45	46
9	0,026	0,012	0,000	0,000	0,006	0,009	41	47
10	0,000	0,000	0,030	0,213	0,163	0,123	45	45

A tabela com o cruzamento das categorias previstas e reais pode ser observada na Tabela 21.

Tabela 21: Resultados da classificação (segundo nível) pela análise de proximidade – documentos sem *stemming*

		Categorias Reais					
		41	42	43	45	46	47
Categorias Previstas	41	4	3	3	2	2	6
	42	19	16	5	1	4	1
	43	0	1	13	3	1	1
	45	0	0	0	15	4	2
	46	0	1	1	2	7	7
	47	0	0	0	3	1	4

Dos 132 documentos apenas **59** são classificados corretamente. A taxa de erro ronda os 55%.

- **Abordagem 2** (documentos COM *stemming*)

Começar por considerar os documentos das empresas e os documentos das categorias, **com** a tarefa *stemmimg*.

```
empresas <- emp_stem
categorias <- cat_stem_N2
```

Considerar apenas os documentos das empresas usados no conjunto de teste para construir a tabela de proximidade.

```
docs_teste <- rownames(dtm.test2_stem)
docs_teste <- as.numeric(as.character(docs_teste))
empresas <- empresas[docs_teste]
```

A Tabela 22 apresenta o *output* parcial da tabela de proximidade entre os documentos das empresas e o descritivo das categorias, documentos aos quais foi aplicada a tarefa *stemming*.

Tabela 22: Output parcial da tabela de proximidade, dos documentos das empresas com os documentos das categorias do primeiro nível, ambos pós *stemming*, com as classificações previstas e reais.

	41	42	43	45	46	47	PREV	REAL
1	0,000	0,033	0,068	0,004	0,068	0,066	46	47
2	0,000	0,034	0,125	0,033	0,043	0,263	47	47
3	0,000	0,020	0,114	0,024	0,040	0,038	43	46
4	0,000	0,000	0,000	0,000	0,027	0,021	46	47
5	0,000	0,000	0,046	0,092	0,197	0,145	46	47
6	0,035	0,108	0,194	0,000	0,036	0,029	43	46
7	0,000	0,000	0,045	0,032	0,018	0,083	47	46
8	0,000	0,053	0,004	0,045	0,048	0,038	42	46
9	0,026	0,011	0,019	0,000	0,007	0,124	47	47
10	0,000	0,040	0,035	0,208	0,169	0,125	45	45

A tabela com o cruzamento das categorias previstas e reais pode ser visualizado na Tabela 23.

Tabela 23: Resultados da classificação (segundo nível) pela análise de proximidade – documentos com *stemming*

		Categorias Reais					
		41	42	43	45	46	47
Categorias Previstas	41	7	3	2	0	3	4
	42	9	15	4	0	1	1
	43	5	5	14	7	5	6
	45	0	0	0	14	2	1
	46	0	0	1	2	3	4
	47	0	0	2	1	4	6

Dos 131 documentos apenas **59** são classificados corretamente. A taxa de erro situa-se nos 55%. Esta abordagem, considerando os documentos pós-*stemming*, revela resultados idênticos aos da abordagem anterior (sem *stemming*).

4.3. Discussão dos Resultados

Neste subcapítulo são discutidos todos os resultados obtidos nos subcapítulos 4.1. e 4.2. No subcapítulo 4.1., foi avaliada a performance dos classificadores considerados neste estudo, variando o tipo de classificação (local por nó pai e local por nível) e o tipo de documentos (sem *stemming* e com *stemming*).

Na classificação no primeiro nível, os resultados nas duas abordagens de classificação são iguais, ou seja, os conjuntos de treino e teste são os mesmos e são constituídos por exemplos de documentos das categorias 'F', 'G' e 'OUTRA'. Em relação ao tipo de documentos, os conjuntos de treino e teste são os mesmos, no entanto existem diferenças a nível da classificação dos documentos nas categorias. Os classificadores que usaram como exemplos de treino, documentos sem *stemming*, obtiveram no geral, resultados ligeiramente melhores, com exceção dos algoritmos árvores de decisão e SVM. Neste nível de classificação, as medidas de avaliação apontam o algoritmo *Naive Bayes* como o melhor classificador nos dois tipos de documentos. Este classificador apresenta uma taxa de acerto de 81.15% na classificação de documentos sem *stemming* e uma taxa de acerto de 80.74% na classificação de documentos com *stemming*, resultados considerados bastante bons. Os resultados obtidos com o classificador SVM e SVM Linear são ligeiramente piores, mesmo assim são considerados bons, as medidas microF1 variam entre os 76% e os 79%.

Na classificação no segundo nível, o tipo de classificação (local por nó pai ou local por nível) implica diferenças na constituição dos conjuntos de treino e teste o que conduz a resultados diferentes.

Na abordagem classificação local por nó pai, são construídos dois classificadores, um para classificar os descendentes de 'F' e outro para classificar os descendentes de 'G'. O

classificador que apresentou melhores resultados na classificação dos descendentes de ‘F’ foi o algoritmo SVM Linear, no caso de documentos sem *stemming*, com uma medida microF1 de 64% e o algoritmo *Naive Bayes*, no caso de documentos com *stemming*, com uma medida microFI de 61%. Já para classificar os descendentes de ‘G’, o algoritmo *Naive Bayes* revelou-se novamente o melhor classificador em ambos os tipos de documentos. No caso dos documentos sem *stemming*, apresentou uma de erro de 19.70% e em documentos com *stemming* essa medida cifrou-se nos 20.31%.

Na abordagem classificação local por nível foi construído um classificador para atribuir os documentos às categorias ‘41’, ‘42’, ‘43’, ‘45’, ‘46’ ou ‘47’. Dos classificadores em estudo, o que obteve a melhor performance foi o *Naive Bayes*. Nos dois tipos de documentos que considerou como exemplos de treino a medida microF1 ronda os 67%, sendo que a taxa de erro é ligeiramente inferior no caso dos documentos sem *stemming*.

A classificação local por nó pai atinge melhores resultados para as categorias ‘45’, ‘46’ e ‘47’ (obtendo uma medida F1 de 0.93, 0.69 e 0.74, respetivamente) do que a classificação local por nível (com medida F1 de 0.90, 0.59 e 0.68). No caso dos resultados obtidos com o classificador SVM Linear para as categorias ‘41’, ‘42’ e ‘43’, a medida F1 é de 0.56, 0.56 e 0.79, respetivamente, na classificação local por nó pai, enquanto a medida F1 obtida pelo classificador *Naive Bayes*, na classificação local por nível, é de 0.68, 0.58 e 0.55, para as categorias ‘41’, ‘42’ e ‘43’, ou seja, na classificação dos descendentes de ‘F’ não existe um tipo de classificação que sobressaia face ao outro, no entanto a medida F1 nas categorias ‘41’ e ‘42’ é superior (0.68>0.56 e 0.58>0.56) no caso da classificação local por nível.

Na análise de proximidade entre os documentos das empresas com os documentos das categorias ‘F’, ‘G’ e ‘OUTRA’, a taxa de documentos corretamente classificados foi de aproximadamente 60%, no caso de considerar todos os documentos sem a tarefa de pré-processamento *stemming* e de aproximadamente 54% no caso de considerar todos os documentos com *stemming*.

Na abordagem que analisa a proximidade entre os documentos das empresas com os documentos das categorias ‘41’, ‘42’, ‘43’, ‘45’, ‘46’ e ‘47’, a taxa de documentos

corretamente classificados foi de aproximadamente 55%, considerando os dois tipos de documentos (sem e com *stemming*).

Portanto, a análise de similaridade não revelou ganhos face ao método de classificação.

CAPÍTULO 5
Conclusões

O projeto abordou a utilização de técnicas de *data mining* / *text mining*, em particular métodos de classificação hierárquica de documentos, com o objetivo de classificar os documentos em categorias pré-definidas.

A coleção de documentos utilizada neste estudo é constituída por 800 documentos, com o descritivo da atividade económica das empresas obtido a partir das respetivas páginas na Internet . As categorias correspondem a algumas Secções e Divisões da CAE-Rev.3, ou seja, as categorias estão estruturadas hierarquicamente.

A classificação de documentos foi efetuada considerando dois métodos hierárquicos, a classificação local por nó pai e a classificação local por nível. Os classificadores construídos treinaram com exemplos de documentos de dois tipos, com e sem aplicação da tarefa de pré-processamento *stemming*.

Para além de métodos de classificação, foi analisada a proximidade entre documentos do descritivo das empresas com os documentos do descritivo das categorias. Numa primeira abordagem, consideraram-se os documentos das empresas do conjunto de teste no primeiro nível e os documentos das categorias do mesmo nível (‘F’, ‘G’ e ‘OUTRA’). Numa segunda abordagem, consideraram-se os documentos das empresas do conjunto de teste no segundo nível e os documentos das categorias do mesmo nível (‘41’, ‘42’, ‘43’, ’45’, ‘46’ e ‘47’). Em ambas as situação foram comparados os resultados com documentos sem e com *stemming*.

O método de seleção do conjunto de treino e do conjunto de teste escolhido tem a vantagem de independência dos exemplos selecionados e da redução de tempo de computação. No entanto, os resultados obtidos por este método originam conclusões consideradas preliminares, terão que ser confirmadas noutros estudos.

Na seleção de características relevantes, o valor mínimo de informação de cada termo considerado em cada uma das situações apresentadas, deveu-se ao valor que melhores

resultados apresentou. Assim, na classificação do primeiro nível, consideraram-se termos com o mínimo de informação de 0.005 e na classificação do segundo nível foram considerados os termos com um ganho mínimo de 0.1.

Como principais resultados deste trabalho destacam-se a boa performance do classificador construído para o primeiro nível, o algoritmo *Naive Bayes*, com uma medida micro F1 de cerca de 81% e também a boa performance do classificador construído para a classificação do segundo nível considerando a abordagem classificação local por nó pai, para classificar os descendentes da categoria 'G', com medida micro F1 de cerca de 80%. Os resultados obtidos por estes classificadores são considerados bastante bons. A classificação dos descendentes do nó 'F' foi considerada melhor através do classificador construído pela abordagem local por nível.

No geral, o algoritmo que apresentou os melhores resultados nos diferentes níveis, considerando os dois métodos de classificação e os dois tipos de documentos, foi o algoritmo *Naive Bayes*, com exceção da classificação local por nó pai, em que o algoritmo, SVM Linear, construído para classificar os descendentes de 'F' apresentou melhores resultados.

A análise de similaridade, como método alternativo de classificação, não revelou melhoria nos resultados face aos obtidos com o método de classificação.

5.1. Considerações Finais

A pretensão de conseguir automatizar o processo de recolha da informação, disponibilizada pelas empresas na respetiva página da *web*, tornou-se no primeiro grande obstáculo deste projeto. A criação da coleção de documentos acabou por ser efetuada manualmente.

Outro obstáculo neste projeto deveu-se ao facto de existirem empresas com diversas atividades. Como o trabalho proposto se baseava numa classificação multi-classe, o código referente à atividade principal foi o considerado para classificar a empresa. No entanto, a nível do descritivo da empresa na Internet, não é fácil de contornar, ou seja,

as empresas descrevem todas as atividades em que atuam e não apenas a atividade principal.

5.2. Trabalhos Futuros

Como trabalho futuro, ficam algumas sugestões:

Refazer o mesmo estudo com outros métodos de seleção dos conjuntos de treino e teste, por exemplo, o método *cross-validation*, em que os exemplos são divididos de forma aleatória em *r* partições mutuamente exclusivas.

Possível automatização do processo de criação da coleção de documentos, ou seja, em vez de obter o descritivo da página na Internet das empresas, considerar, por exemplo, um problema de classificação de documentos *web*.

Considerando que uma empresa pode estar classificada com diferentes códigos CAE-Rev.3., podia ser abordada a classificação multi-rótulo deste problema.

Como tanto a classificação local por nó pai como a classificação local por nível envolve múltiplas classes, é previsível que o uso de estratégia *One-Against-All* possa trazer benefícios. Esta estratégia usa um classificador binário para cada categoria, distinguindo cada categoria das outras. Assim, outra sugestão remete ao uso da estratégia *One-Against-All* no problema de classificação em estudo.

Seria ainda interessante estudar problemas com um maior número de categorias e subcategorias para ver qual é a melhor estratégia. Como é difícil recolher dados, teria que se recorrer a conjuntos de dados recolhidos por outros.

Bibliografia

Awad, M., Khan, L., Thuraisingham, B. e Wang, L. (2009). *Design and Implementation of Data Mining Tools*. Taylor & Francis Group. Boca Raton.

Brazdil, P. (2009). Feature Selection in R. *Slides de ECDI*.

Bramer, M. (2007). *Principles of Data Mining*. Springer. New York.

Dumais, S. e Chen, H. (2000). Hierarchical Classification of Web Content. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, Greece, 24 - 28 July 2000. pp. 256-263.

Faceli, K., Lorena, A. C., Gama, J. e Carvalho, A. C. P. L. F. (2011). *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. LTC. Rio de Janeiro.

Feinerer, I., Hornik, K. e Meyer D. (2008). Text mining infrastructure in R. *Journal of Statistical Software*, **25:** 1–54.

Feinerer, I. (2011, 20 de Fevereiro). Introduction to the tm Package: Text Mining in R. Acedido a 19 de Novembro 2011, em: http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf

Feldman, R. e Sanger, J. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press. New York.

Han, J. e Kamber, M. (2006). *The Data Mining: Concepts and Techniques*. Second Edition, Morgan Kaufmann Publishers. San Francisco.

Honrado, A., Leon, R., O'Donnel, R. e Sinclair, D. (2000). A Word Stemming Algorithm for the Spanish Language. *Proceedings of the Seventh International Symposium on String Processing Information Retrieval*, Spain, 27 – 29 September 2000. pp. 139-145.

Instituto Nacional de Estatística, IP. (2007). *Classificação Portuguesa das Actividades Económicas Rev.3*. INE. Lisboa.

Kantardzic, M. (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. Second Edition, John Wiley & Sons. New Jersey.

Michie, D., Spiegelhalter, D. J. e Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood.

Mihaescu, C. (2011). *Naive-Bayes Classification Algorithm*. Acedido em 15 de Setembro de 2012, no *Web site* da: Universitatea din Craiova, Departamentul de Inginerie Software: http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Companies, Inc.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, **14:**130-137. Acedido em http://www.tartarus.org/~martin/PorterStemmer/def.txt

Press, W. H., Teukolsky, S. A., Vetterling, W. T. e Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*. Third edition. Cambridge University Press. New York.

Russell, I. (1996). *Neural Networks Module*. Acedido em 15 de Setembro de 2012, em http://uhaweb.hartford.edu/compsci/neural-networks-definition.html.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, New York, March 2002. **34:**1-47.

Silla Jr.,C. N. e Freitas, A. A. (2011). A Survey of Hierarchical Classification Across Different Application Domains. *Data Mining and Knowledge Discovery*. **22:**31-72.

Sun, A. e Lim, E.-P. (2000). Hierarchical Text Classification and Evaluation. *Proceedings of the 2001 IEEE International Conference on Data Mining*, USA, November 2001. pp. 521-528.

Tan, P.-N., Steinbach, M. e Kumar, V. (2006). *Introdution to Data Mining*. Pearson Addison-Wesley.

Weiss, S. M., Indurkhya, N. e Zhang, T. (2010). *Fundamentals of Predictive Text Mining.* Springer. London.

Witten, I. H. e Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers. San Francisco.

[1] Informa D&B. "Directório de Todas as Empresas Portuguesas", http://directorio.informadb.pt/, acedido entre Novembro de 2011 e Agosto de 2012.

[2] R. "About R?", http://www.r-project.org/, acedido em 23 de Junho de 2012.

[3] Wikipedia. "Alfabeto latino". http://pt.wikipedia.org/wiki/Alfabeto_latino, acedido em 14 de Abril de 2012.

[4] Wikipedia. "ISO 8859-1". http://pt.wikipedia.org/wiki/ISO_8859-1, acedido em 14 de Abril de 2012.

[5] Wikipedia. "Text mining", http://en.wikipedia.org/wiki/Text_mining, acedido em 1 de Julho de 2012.

ANEXOS

Anexo 1: Lista das Secções e suas relações com as Divisões.

CAE-Rev.3		
SECÇÃO	**DESIGNAÇÃO**	**RELAÇÃO SECÇÃO/DIVISÃO**
A	Agricultura, produção animal, caça, floresta e pesca	01+02+03
B	Indústrias extractivas	05+06+07+08+09
C	Indústrias transformadoras	10+11+12+13+14+15+16+17+18+19+20+21+22+23+24+25+26+27+28+29+30+31+32+33
D	Electricidade, gás, vapor, água quente e fria e ar frio	35
E	Captação, tratamento e distribuição de água; saneamento, gestão de resíduos e despoluição	36+37+38+39
F	Construção	41+42+43
G	Comércio por grosso e a retalho; reparação de veículos automóveis e motociclos	45+46+47
H	Transportes e armazenagem	49+50+51+52+53
I	Alojamento, restauração e similares	55+56
J	Actividades de informação e de comunicação	58+59+60+61+62+63
K	Actividades financeiras e de seguros	64+65+66
L	Actividades imobiliárias	68
M	Actividades de consultoria, científicas, técnicas e similares	69+70+71+72+73+74+75
N	Actividades administrativas e dos serviços de apoio	77+78+79+80+81+82
O	Administração Pública e Defesa; Segurança Social Obrigatória	84
P	Educação	85
Q	Actividades de saúde humana e apoio social	86+87+88
R	Actividades artísticas, de espectáculos, desportivas e recreativas	90+91+92+93
S	Outras actividades de serviços	94+95+96
T	Actividades das famílias empregadoras de pessoal doméstico e actividades de produção das famílias para uso próprio	97+98
U	Actividades dos organismos internacionais e outras instituições extra-territoriais	99

Fonte: INE (2007)

Anexo 2: Código para eliminar o plural de um termo, no caso de existir o seu singular, e juntar a informação de ambos.

```
# Todas as palavras (termos) da matriz de documentos-termos (dtm)
palavras <- as.character(colnames(dtm))

# REGRA 1 --- Palavras que terminam em 's' e a penúltima letra é uma vogal:
pos_plural1 <- which(palavras %in% paste(palavras,'s',sep='')
                & (substr(palavras,nchar(palavras)-1,nchar(palavras)-1) %in% c('ã','a','e','i','o','u')
                 | substr(palavras,nchar(palavras)-2,nchar(palavras)-1) %in% c('ãe'))
                & !substr(palavras,nchar(palavras)-2,nchar(palavras)-1) %in% c('ão','õe'))
# REGRA 2 --- Palavras que terminam em 'es' e a antepenúltima letra pertence a ('r','z','n'):
pos_plural2 <- which(palavras %in% paste(palavras,'es',sep='')
& substr(palavras,nchar(palavras)-2,nchar(palavras)-2) %in% c('r','z','n'))
# REGRA 3 --- Palavras que terminam em 'is' e ao substituir de 'is' por 'l' a palavra termina em ('al','el','ol','ul'):
pos_plural3 <- which(palavras %in% paste(substr(palavras,1,nchar(palavras)-1),'is',sep='')
& paste(substr(palavras,1,nchar(palavras)-2),'l',sep='') %in% palavras
& paste(substr(palavras,nchar(palavras)-2,nchar(palavras)-2),'l',sep='') %in% c('al','el','ol','ul'))
# REGRA 4 --- Palavras que terminam em 'is' e ao substituir 's' por 'l' a palavra termina em 'il':
pos_plural4 <- which(palavras %in% paste(substr(palavras,1,nchar(palavras)-1),'s',sep='')
          & paste(substr(palavras,1,nchar(palavras)-1),'l',sep='') %in% palavras
         & paste(substr(palavras,nchar(palavras)-1,nchar(palavras)-1),'l',sep='') %in% c('il'))
# REGRA 5 --- Palavras que terminam em 'ão':
pos_plural5 <- which(palavras %in% paste(substr(palavras,1,nchar(palavras)-1),'s',sep='')
& paste(substr(palavras,1,nchar(palavras)-3),'ão',sep='') %in% palavras
& substr(palavras,nchar(palavras)-2,nchar(palavras)-1) %in% c('ão','ãe','õe')
& palavras!='mões' )
pos_plural5 <- pos_plural5 [!pos_plural5 %in% pos_plural1]
pos_plurais <- c(pos_plural1, pos_plural2, pos_plural3, pos_plural4, pos_plural5)

plurais <- unique(palavras[pos_plurais])
singular <- unique(ifelse(plurais %in% palavras[pos_plural1], substr(plurais,1,nchar(plurais)-1),
```

```
    ifelse(plurais %in% palavras[pos_plural2], substr(plurais,1,nchar(plurais)-2),
    ifelse(plurais %in% palavras[pos_plural3], paste(substr(plurais,1,nchar(plurais)-2),'l',sep=''),
    ifelse(plurais %in% palavras[pos_plural4], paste(substr(plurais,1,nchar(plurais)-1),'l',sep=''),
    ifelse(plurais %in% palavras[pos_plural5], paste(substr(plurais,1,nchar(plurais)-
3),'ão',sep=''),'')))))

vector_col <- NULL
for(i in 1:length(palavras))
   {
    if(palavras[i] %in% singular) {
    j <- which(palavras %in% paste(palavras[i],'s',sep='') &
         substr(palavras[i],nchar(palavras[i]),nchar(palavras[i])) %in% c('ã','a','e','i','o','u')
&!substr(palavras[i],nchar(palavras[i])-1,nchar(palavras[i])) %in% c('ão','õe')
         | palavras %in% paste(palavras[i],'es',sep='')
& substr(palavras[i],nchar(palavras[i]),nchar(palavras[i])) %in% c('r','z','n')
         |palavras %in% paste(substr(palavras[i],1,nchar(palavras[i])-1),'is',sep='')
& substr(palavras[i],nchar(palavras[i])-1,nchar(palavras[i])) %in% c('al','el','ol','ul')
         |palavras %in% paste(substr(palavras[i],1,nchar(palavras[i])-1),'s',sep='')
& substr(palavras[i],nchar(palavras[i])-1,nchar(palavras[i])) %in% c('il')
         |palavras %in% c(paste(substr(palavras[i],1,nchar(palavras[i])-2),'ãos',sep=''),
                    paste(substr(palavras[i],1,nchar(palavras[i])-2),'ães',sep=''),
                    paste(substr(palavras[i],1,nchar(palavras[i])-2),'ões',sep='')) &
                   substr(palavras[i],nchar(palavras[i])-1,nchar(palavras[i])) %in% c('ão')
          )

    if(length(j)!=1) j<-j[2]
    dtm[,i] <- dtm[,i]+ dtm[,j]
    vector_col <- c(vector_col,j)
                        }
   }
dtm <- dtm[,-vector_col]
```

Anexo 3: Comparação de algumas funções de *stemming* do R, numa pequena amostra de palavras usadas neste estudo.

Palavras	Package 'tm'	Package 'Rstem'	Package 'Snowball'	Package 'Rweka'	Package 'Rweka'
	stemDocument()	wordStem ()	SnowballStemmer ()	IteratedLovinsStemmer()	LovinsStemmer()
água	águ	águ	água	águ	águ
ano	ano	ano	ano	an	an
apenas	apen	apen	apena	apen	apen
ardósia	ardós	ardós	ardósia	ardó	ardós
artigos	artig	artig	artigo	artig	artigo
cae	cae	cae	cae	ca	ca
cal	cal	cal	cal	cal	cal
cerca	cerc	cerc	cerca	cerc	cerc
destina	destin	destin	destina	destin	destin
distribuição	distribuiçã	distribuiçã	distribuição	distribuiçã	distribuiçã
dizem	diz	diz	dizem	dizem	dizem
exclusivamente	exclus	exclus	exclusivament	exclusivam	exclusivament
exportações	export	export	exportaçõ	exportaçõ	exportaçõ
extração	extraçã	extraçã	extração	extraçã	extraçã
extracção	extracçã	extracçã	extracção	extracçã	extracçã
extraída	extraíd	extraíd	extraída	extraíd	extraíd
extraído	extraíd	extraíd	extraído	extraíd	extraíd
fábrica	fábric	fábric	fábrica	fábr	fábric
fabricação	fabric	fabric	fabricação	fabricaçã	fabricaçã
fabricado	fabric	fabric	fabricado	fabricad	fabricad
fabrico	fabric	fabric	fabrico	fabr	fabric
homogéneo	homogén	homogén	homogéneo	homogén	homogéne
importação	import	import	importação	importaçã	importaçã
industriais	industri	industri	industriai	industr	industria
mármore	mármor	mármor	mármore	mármor	mármor
material	material	material	materi	mater	mater
mercado	merc	merc	mercado	mercad	mercad
milhões	milhõ	milhõ	milhõ	milhõ	milhõ
nacional	nacional	nacional	nacion	nac	nac
pedra	pedr	pedr	pedra	pedr	pedr
pedreiras	pedreir	pedreir	pedreira	pedreir	pedreir
processadas	process	process	processada	processad	processad
produto	produt	produt	produto	produt	produt
qualidade	qualidad	qualidad	qualidad	qualidad	qualidad
quase	quas	quas	quas	qu	quas
relativos	relat	relat	relativo	relativ	relativo
representando	represent	represent	representando	representand	representand
respeito	respeit	respeit	respeito	respeit	respeit
rocha	roch	roch	rocha	roch	roch
rochas	roch	roch	rocha	roch	roch
similares	simil	simil	similar	simil	similar
subsector	subsector	subsector	subsector	subsect	subsect
subsetor	subsetor	subsetor	subsetor	sub	subses
toneladas	tonel	tonel	tonelada	tonelad	tonelad
total	total	total	total	tot	tot
utilizada	utiliz	utiliz	utilizada	utilizad	utilizad

Anexo 4: Ficheiro parcial das empresas consideradas neste estudo com os respetivos códigos CAE_Rev.3

NOME_EMPRESA	URL	Secção	Divisão	Grupo	Classe	Subclasse
PIONEER HI-BRED SEMENTES DE PORTUGAL, S.	http://www.pioneer.com/portugal/	G	46	462	4621	46214
CONTINENTE - HIPERMERCADOS, S.A.	http://www.sonae.pt/gca/index.php?id=66	G	47	471	4711	47111
ZIPPY - COMÉRCIO E DISTRIBUIÇÃO, S.A.	http://www.sonae.pt/pt/marcas/zippy/	G	47	477	4771	47712
TRANSPORTES AEREOS PORTUGUESES, S.A.	http://www.tapportugal.com/Info/pt/SobreaT	H	51	511	5110	51100
PT COMUNICAÇÕES, S.A.	http://www.ptcom.pt/PTResidencial2/Tabs/So	J	61	611	6110	61100
EDP DISTRIBUIÇÃO - ENERGIA, S.A.	http://www.edp.pt/pt/aedp/unidadesdenegoc	D	35	351	3513	35130
PINGO DOCE - DISTRIBUIÇÃO ALIMENTAR, S.A	http://www.pingodoce.pt/artigo.aspx?sid=696	G	47	471	4711	47111
TMN - TELECOMUNICAÇÕES MÓVEIS NACIONA	http://www.tmn.pt/portal/site/tmn/menuiten	J	61	611	6110	61100
EDP COMERCIAL - COMERCIALIZAÇÃO DE ENER	http://www.edp.pt/pt/aedp/unidadesdenegoc	D	35	351	3514	35140
VODAFONE PORTUGAL - COMUNICAÇÕES PES	http://www.vodafone.pt/main/A+Vodafone/P	J	61	611	6110	61100
CTT EXPRESSO - SERVIÇOS POSTAIS E LOGÍSTIC	http://www.cttexpresso.pt/fecewcm/wcmserv	H	53	532	5320	53200
ZON LUSOMUNDO CINEMAS, S.A.	http://www.zon.pt/microsites/investidores/ge	J	59	591	5914	59140
LACTOGAL - PRODUTOS ALIMENTARES, S.A.	http://www.lactogal.pt/presentationlayer/ctex	C	10	105	1051	10510
MOTA-ENGIL, ENGENHARIA E CONSTRUÇÃO, S	http://www.mota-engil.pt/AreaDetail.aspx?co	F	42	421	4211	42110
ALLIANCE HEALTHCARE, S.A.	http://www.alliance-healthcare.pt/a-atividade	G	46	464	4646	46460
ZON LUSOMUNDO AUDIOVISUAIS, S.A.	http://www.zon.pt/microsites/investidores/ge	J	59	591	5913	59130
ZON CONTEÚDOS - ACTIVIDADE DE TELEVISÃO	http://www.zon.pt/microsites/investidores/ge	J	60	602	6020	60200
PORTUCEL - EMPRESA PRODUTORA DE PASTA	http://www.portucelsoporcel.com/pt/group/n	C	17	171	1711	17110
ZON TV CABO PORTUGAL, S.A.	http://www.zon.pt/microsites/investidores/ge	J	61	611	6110	61100
HUSSEL IBÉRIA - CHOCOLATES E CONFEITARIA,	http://www.jeronimomartins.com/negocios/se	G	47	472	4724	47240
BRISA - AUTO-ESTRADAS DE PORTUGAL, S.A.	http://www.brisa.pt/PresentationLayer/conte	H	52	522	5221	52211
OCP-PORTUGAL - PRODUTOS FARMACEUTICOS	http://www.ocp.pt/pages/ocp_apresentacao.p	G	46	464	4646	46460
WORTEN - EQUIPAMENTOS PARA O LAR, S.A.	http://institucional.worten.pt/	G	47	471	4719	47191
VICTOR GUEDES - INDÚSTRIA E COMÉRCIO, S.A	http://www.jeronimomartins.com/negocios/in	C	10	104	1041	10412
VANIBRU - COMÉRCIO DE PRODUTOS ALIMENT	http://www.vanibru.pt/	G	46	463	4639	46390
JMV - PRODUTOS HOSPITALARES, LDA	http://www.jmv.com.pt/index.php?action=his	G	46	464	4649	46494
SOMANTIS - MERCEARIA, UNIPESSOAL, LDA	http://www.chasdomundo.pt/	G	47	471	4711	47112
AGROSPORT - PRODUTOS, EQUIPAMENTOS E T	http://www.agrosport.pt/pt/pagina/2/agrospo	G	46	466	4663	46630
PRAMADEIRA - MÁQUINAS E FERRAMENTAS, S	http://www.pramadeira.net/dynamicdata/Hist	G	46	466	4669	46690
INTERFER - MALCATO, FRANCISCO ANTÓNIO	http://www.interfer.pt/	G	46	467	4674	46740
ESCOLAS CAMBRIDGE, S.A.	http://www.cambridge.pt/PT/escola/	P	85	855	8559	85592
TAC CREATIVE MANUFACTURING UNIPESSOAL	http://www.tac.pt/servicos.html	C	14	141	1413	14132
EMBA - COMÉRCIO E INDÚSTRIA DE EMBALAGE	http://www.emba.pt/produtos.html	C	17	172	1721	17212
OLIVEIRA & IRMÃO, S.A.	http://www.oli.pt/scid/olweb13/defaultCateg	C	22	222	2223	22230
BRAMP - METAIS E POLÍMEROS DE BRAGA, LDA	http://www.bramp.pt/bramp/	C	22	222	2229	22292
PLACE FORMAÇÃO - EDUCAÇÃO E FORMAÇÃO	http://www.grupoplace.com/home.html	P	85	855	8559	85591
LUSICAL - COMPANHIA LUSITANA DE CAL, S.A.	http://www.lusical.pt/	C	23	235	2352	23521
C.S.C.PORTUGUESA - CALDEIRAS ESPECIAIS PA	http://www.csc-caldeiras.com/	C	25	253	2530	25300
INFORMA D & B (SERVIÇOS DE GESTÃO DE EMP	https://www.informadb.pt/idb/publico/quems	N	82	829	8299	82990
TEIXEIRA DUARTE - ENGENHARIA E CONSTRUÇ	http://www.teixeiraduarte.pt/setores-de-ativi	F	42	429	4299	42990
IDADE VIRTUAL - FORMAÇÃO INFORMATICA, L	http://www.idadevirtual.pt/index.php?option	P	85	855	8559	85591
TABAQUEIRA - EMPRESA INDUSTRIAL DE TABA	http://www.pmi.com/pt_pt/about_us/pages/a	C	12	120	1200	12000
PT - SISTEMAS DE INFORMAÇÃO, S.A.	http://www.ptsi.pt/PTSI/quem_somos.html	J	62	620	6202	62020
GO FLAG, S.A.	http://www.flag.pt/pages/quem_somos.asp	P	85	855	8559	85591
AUTO DIESEL - PROGRESSO DE ALENQUER, LDA	http://www.auto-diesel.com/index.php?optio	G	45	452	4520	45200
TURBOGÁS - PRODUTORA ENERGÉTICA, S.A.	http://www.turbogas.pt/gca/index.php?id=95	D	35	351	3511	35112
AUTOCOOPE - COOPERATIVA DE TÁXIS DE LISB	http://www.cooptaxis.pt/cooptaxis/index.htm	H	49	493	4932	49320
TRANSPORTES MATOS & FILHOS LDA	http://www.transportesmatos.pt/	H	49	494	4941	49410
LIDERCISTER - TRANSPORTES DE PULVERULENT	http://www.lidercister.pt/?lang=pt&op=empre	H	49	494	4941	49410
ANTÓNIO MOCHO, LDA	http://www.granidias.pt/	B	8	81	811	8112
GRANITOS DE MACEIRA, SA	http://www.granitos-maceira.com/po/index.p	B	8	81	811	8112
FARPEDRA - EXPLORAÇÃO DE PEDREIRAS, LDA	http://www.farpedra.com/index.php?m=page	B	8	81	811	8113
GRANIDERA-GRANITOS DE PEDRA D'ERA, S.A.	http://www.granidera.pt/	B	8	81	811	8112
SOCIEDADE DAS PEDREIRAS DO MARCO LDA	http://www.portuguesegranite.com/index_pt.	B	8	81	811	8112
MOTAMINERAL, MINERAIS INDUSTRIAIS, SA	http://www.mota-sc.com/main_pt/motaminer	B	8	81	812	8122
FELMICA - MINERAIS INDUSTRIAIS, S.A.	http://www.mota-sc.com/main_pt/felmica.sht	B	8	89	899	8991
R & G - ROGRANIT GRALPE GRANITOS, LDA	http://www.rg-granitos.pt/pt/home.html	B	8	81	811	8112
SORGILA-SOCIEDADE DE ARGILAS, SA	http://www.sorgila.com/site/	B	8	81	812	8121
LENA AGREGADOS - COMÉRCIO DE AGREGADO	http://www.lenaagregados.pt/agregados.php	B	8	81	812	8121
SECIL PREBETÃO - PREFABRICADOS DE BETÃO,	http://www.secilprebetao.pt/gca/?id=51	C	23	236	2361	23610

Anexo 5: Função info() que calcula a informação do termo *i* e função find.info.terms() que seleciona os termos com maior informação

```
info <- function(x){
                    inf <- 0
                     sumx <- sum(x)
                     for (i in x)
                               {
                               p_i <- i/sumx
                                inf_i <- (p_i)*log2(p_i)
                                if(is.na(inf_i)) inf_i <- 0
                                inf <- inf-inf_i
                                 }
                        return(inf)
                     }


info.terms <- vector()
find.info.terms <- function(dtm.train,min.info)
  {
  ix.class <- ncol(dtm.train)
  default.info <- info(table(dtm.train[,ix.class]))
  cat("default.info: ", default.info, "\n")
  n.atr <- ncol(dtm.train)-1
  n.info.terms <- 0
  info.term.ixs <- vector()
  col.names <- names(dtm.train)
   for (atri in 1:n.atr)
     {
     if(sum(dtm.train[,atri]) > 0)
       {
       no.dif.atr.val <- length(table(dtm.train[,atri]))
       atr.class.table <- table(dtm.train[,atri],dtm.train[,ix.class])
```

```
        n.rows <- nrow(dtm.train)
        atr.info <- 0
           for (atr.val in 1:no.dif.atr.val)
              {
              atr.peso <- sum(atr.class.table[atr.val,]) / n.rows
              atr.info1 <- atr.peso*info(atr.class.table[atr.val,])
              atr.info <- atr.info+atr.info1
              }
        info.gain <- default.info-atr.info
            if(info.gain > min.info)
              {
              info.term.ixs[n.info.terms] <- atri
              n.info.terms <- n.info.terms+1
              }
        }
      }
   cat("\n", "Vão ser mantidos ", n.info.terms, " atributos: ", "\n")
   cat(col.names[info.term.ixs[1:5]], " ...  ")
   cat(col.names[info.term.ixs[n.info.terms-1]],"\n")
   cat("Vão ser eliminados ", n.atr-n.info.terms, " atributos", "\n")
return(col.names[info.term.ixs])
}
```

Fonte: Brazdil (2009)

Anexo 6: Documento com a descrição da categoria 'F'

Construção.
Promoção imobiliária (desenvolvimento de projectos de edifícios); construção de edifícios.
Promoção imobiliária (desenvolvimento de projectos de edifícios).
Construção de edifícios (residenciais e não residenciais).
Engenharia civil.
Construção de estradas, pontes, túneis, pistas de aeroportos e vias férreas.
Construção de estradas e pistas de aeroportos.
Construção de vias férreas.
Construção de pontes e túneis.
Construção de redes de transporte de águas, de esgotos, de distribuição de energia, de telecomunicações e de outras redes.
Construção de redes de transporte de águas, de esgotos e de outros fluidos.
Construção de redes de transporte e distribuição de electricidade e redes de telecomunicações.
Construção de outras obras de engenharia civil.
Engenharia hidráulica.
Construção de outras obras de engenharia civil, n. e.
Actividades especializadas de construção.
Demolição e preparação dos locais de construção.
Demolição.
Preparação dos locais de construção.
Perfurações e sondagens.
Instalação eléctrica, de canalizações, de climatização e outras instalações.
Instalação eléctrica.
Instalação de canalizações e de climatização.
Instalação de canalizações.
Instalação de climatização.
Outras instalações em construções.
Actividades de acabamento em edifícios.
Estucagem.
Montagem de trabalhos de carpintaria e de caixilharia.
Revestimento de pavimentos e de paredes.
Pintura e colocação de vidros.
Outras actividades de acabamento em edifícios.
Outras actividades especializadas de construção.
Actividades de colocação de coberturas.
Outras actividades especializadas de construção, n. e.
Aluguer de equipamento de construção e de demolição, com operador.
Outras actividades especializadas de construção diversas, n. e.

Anexo 7: Função Evaluating_Classifer() que calcula as medidas de avaliação dos algoritmos

```
Evaluating_Classifer <- function (conf.mx,n_cl=length(unique(class.train))) {
        if(n_cl==2)
                {
                tp <- conf.mx[1,1]
                fp <- conf.mx[2,1]
                fn <- conf.mx[1,2]
                tn <- conf.mx[2,2]

         error.rate <- (fp+fn) / (tp+tn+fp+fn)
         precision1 = tp / (tp+fp)
         recall1 = tp / (tp+fn)
        f1_1 = (2 * precision1 * recall1) / (precision1 + recall1)
         precision2 = tn / (tn+fn)
        recall2 = tn / (tn+fp)
        f1_2 = (2 * precision2 * recall2) / (precision2 + recall2)
         macrof = ((precision1 + precision2)*(recall1+recall2)/2) / ((precision1 + precision2) / 2
        +(recall1 + recall2) / 2)
         microf1 = sum(diag(conf.mx))/sum(conf.mx)
                }else{
        tp <- vector()
        fp <- vector()
        fn <- vector()
        tn <- vector()
        precision <- vector()
        recall <- vector()
        f1 <- vector()

                for(i in 1:n_cl){
                                tp[i] <- conf.mx[i,i]
```

```
                                    fp[i] <- sum(conf.mx[,i])-conf.mx[i,i]
                                    fn[i] <- sum(conf.mx[i,])-conf.mx[i,i]
                                    tn[i] <- sum(conf.mx)-tp[i]-fp[i]-fn[i]

  error.rate <- 100*(sum(conf.mx)-sum(diag(conf.mx)))/sum(conf.mx)
  precision[i] = tp[i] / (tp[i]+fp[i])
  recall[i] = tp[i] / (tp[i]+fn[i])
  f1[i] = (2 * precision[i] * recall[i]) / (precision[i] + recall[i])
  macrof1 = 2*(sum(precision)/n_cl*sum(recall)/n_cl)/(sum(precision)/n_cl+(sum(recall))/n_cl)
  microf1 =
2*(sum(tp)/sum(tp+fp)*sum(tp)/sum(tp+fn))/(sum(tp)/sum(tp+fp)+sum(tp)/sum(tp+fn))
                                          }
              }
              cat("Taxa de Erro:", round(error.rate,3), sep="", "\n")
    for(i in 1:n_cl){
        cat("Precisão",i,":", round(precision[i],3), sep="", "\n")
        cat("Recall",i,":", round(recall[i],3), sep="", "\n")
        cat("F1_",i,":", round(f1[i],3), sep="", "\n")
        }
        cat("Macro F1:", round(macrof1,3), sep="", "\n")
        cat("Micro F1:", round(microf1,3), sep="", "\n")
    return(c(error.rate,precision,recall,f1,macrof1,microf1))
}
```

Anexo 8: Código para comparar as categorias e para efetuar as contagens da similaridade no R

```
# RESULTADOS da SIMILARIDADE

cl <- NULL
 for (i in 1:dim(simil_cos)[1])
      {
      sim_maxima <- names(simil_cos[i,][simil_cos[i,]==apply(simil_cos,1,max)[i]])
      cl <- rbind(cl, sim_maxima)
      }
rownames(cl) <- 1:dim(simil_cos)[1]
colnames(cl) <- 'SECÇÃO'
cl <- substr(cl,1,1)


# CONTAGEM CORRECTOS

emp_CAE <- empresas_CAE[docs_teste,]
a1<-cbind(cl,emp_CAE$SEC)
a1 <- as.data.frame(a1)
names(a1) <- c('PREV','REAL')
a1$PREV <- ifelse(a1$PREV=='F','F',
        ifelse(a1$PREV=='G','G', 'OUTRA'))
a1$REAL <- ifelse(a1$REAL=='F','F',
        ifelse(a1$REAL=='G','G', 'OUTRA'))
sum(a1$PREV==a1$REAL)


tab_out <- table(a1$PREV,a1$REAL)
```